ENSIM PRO - WINDOWS

Ensim Pro 10.3.0 for Windows 2003 Web Services API Guide

ensim

Published: 3 May 2007

CORPORATE HEADQUARTERS

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

www.ensim.com

Contents

Document revision history

There have been no changes to this document since its initial publication. For information about new features in this release, see the What's New document on the Ensim support site, http://support.ensim.com.

CHAPTER 1

About this guide

Introduction

This document provides an overview of Web Services API for Ensim Pro for Windows.

The APIs are grouped under three types of categories.

- Service Provider related actions
- Service Provider/Reseller related actions
- Site administrator related actions

Note: This guide is updated as new information becomes available. Before you begin, be sure to check the Ensim Pro for Windows documentation section of the Ensim Support Site, http://www.ensim.com/support, to make sure you have the latest version of the guide. The date on the title page indicates the version; changes, if any, are recorded in the document revision history (page 5).

In this section:

Intended audience

The audience for the document is primarily engineers who are integrating Ensim Pro into an existing infrastructure.

The target audiences for Ensim Pro for Windows Web Services are:

- Service providers and resellers who want to automate common provisioning tasks
- Third-party tool developers who want to develop tools for provisioning Ensim Pro for Windows Web services

Related documentation

For a list of related documentation, see the Ensim Pro for Windows release notes. Documentation is also available in the Ensim Pro for Windows 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.

Document conventions

Throughout this guide, specific fonts are used to identify user input, computer code, and user interface elements. The following table lists conventions used in this guide.

Table 1. Document conventions

Convention	How it is used
Bold text, serif font	Used for information you type as well as for file names, path names, and CD names. *Example:* On the command line, type **continue.**
Italic text	Used for column names, field names, window names, and book titles. *Example:* The *Select Installation Folder* window is displayed.
Bold, italic text enclosed in angle brackets	Used for variables you replace with the appropriate information. *Example:* Type ***<server_name>*** where ***<server_name>*** is the IP address or host name of your server.
Bold text, sans serif font	Used for items you click or press, such as buttons, menus, and command keys. *Example:* Click Start > Settings > Control Panel.
`Non-proportional font`	Used for system messages, screen text, and code examples. *Example:* The following message is displayed: `The server has been added.`

Note: In addition, key information is sometimes displayed using special headings and formats, such as this one, to make it stand out from regular text.

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_sln.asp 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.

CHAPTER 2

Web Services API calls

Introduction

Ensim Pro for Windows Web Services are programming interfaces that are part of the Ensim Pro Business Systems Integration Framework. They enable you to automate common tasks such as provisioning and managing sites.

Ensim Pro Web Services use WSDL for description, SOAP as the messaging protocol, and HTTP as the transport protocol. Installed on Ensim Pro servers as separate ASP.NET applications, Ensim Pro Web Services connect to the provisioning engine application running on the Ensim Pro servers.

This section describes the Web Services API calls that you can use on the Ensim Pro for Windows server to:

- Create, edit, view, and delete resellers, sites, users, or user templates
- Export and import site information
- Create, edit, view and delete database tools
- Create and delete Microsoft SQL Server 2005 databases
- Update the Ensim Pro for Windows license
- Set the encoding type for email messages

In this section

URL for the Web Services

Use an Internet-connected computer to access the Ensim Pro for Windows Web Services WSDL on the Ensim Pro server. For the Ensim Pro for Windows Web Services, use the following URL:

http://<*myserver.com*>/epwhostingws/hostingws.asmx

where, <***myserver.com***> is the IP address of the Ensim Pro for Windows server, where the Web Services programming interface is running.

Getting started

This section explains how to create a C# client for Ensim Pro Web Services using Visual Studio .NET. The steps in this tutorial assume that the Web Services application is installed on a server called *myserver.myisp.com*. When using this example, replace this server name with the name of your own Ensim Pro server.

Creating a simple C# client

Verify that the Web Service is running

To verify that the Web service is running, open a Web browser and go to the following URL to get a list of supported operations:

http://myserver.myisp.com/epwhostingws/hostingws.asmx

Access the WSDL for the Web Services

To access the WSDL for the service, open a Web browser and type the following URL:
http://myserver.myisp.com/epwhostingws/hostingws.asmx?WSDL

I. Create a new client application in Visual Studio .NET

1. Open Visual Studio .NET
2. Create a new Visual C# Project using the **Console Application** template. In this tutorial the application is called *epwwstest*.

> **Note**: You can access Web Services from all kinds of different applications. In this example, the Console Application template was selected for its simplicity.

II. Add the Web reference to your project

1. In the **Solution Explorer** of your project, right-click and select **Add Web Reference**.
2. In the **URL** box, type **http://myserver.myisp.com/epwhostingws/hostingws.asmx.**
3. In the **Web reference name** box, type **provisioningservice**. This is the name used internally to reference the proxy classes.
4. Click **Add Reference**.
5. At the top of your code add the reference as follows:

```
using System;
using epwwstest.provisioningservice;
```

where, `epwwstest` is the name of the default namespace (usually the name of the application), and `provisioningservice` is the Web reference name selected.

III. Add the code to provision a site

Note: When using the code below, change the values, such as username and password, to match your setup.

1 In your main function, add the following code:

```
// Part 1: Set up connection information for the Web Service (page 12)
  HostingWS w = new HostingWS();
  w.Url = "http://myserver.myisp.com/epwhostingws/hostingws.asmx";
  w.Timeout = 100000;  //time out in milliseconds, -1 implies infinite
// Part 2: Create the site object for the new site (page 12)
  Site site = new Site();
  site.DisplayName = "mysite.com";      // Provide required site name
  site.Name = "mysite.com";
  site.ShortName = "mysite.com";
  site.TemplateName = "Default Template";
  // Specify the template using which the site has to be created
  site.ContactEmail = "blackhole@ensim.com";
  site.AdminUserName = "admin";
  site.AdminDisplayName = "Administrator";
  site.AdminPassword = "ensim123";
  //Provide a password or the provisioning will fail
  site.AdminEmail = "user@example1.com";
  //Specify site administrator email address
  site.IpBased = true;
  // Specify if site will be IP based
  site.IpSelection = "auto";
  // Specify IP address manually or 'auto' for automatic IP allocation
// Part 3: Call the Web Service (page 12)
  w.AddSite(site, "admin", "ensim123");
  //Where 'admin' is the service provider/Reseller username and
  //'ensim123' is the password
  Console.WriteLine( "Created site:mysite.com");
```

2 Build the project and run it.

3 Log in to the service provider control panel, and verify that the site appears on the list of sites.

Code details

This section provides details about the code example shown in Step III (page 11).

Part 1: Set up connection information for the Web service

In this step you provide the connection information, which is passed to the Web Service in all API calls made in the header for the request.

Part 2: Create the site object for the new site

This step creates the site object. This object sets the short name, name, and other required information for the site. When working with objects like this, you are simply constructing proxy objects. At this point nothing is being sent to the Web Service.

Part 3: Call the Web Service

This is the only place that the Web Service is called. The Provisioning Service (w) is the Web Services object. If the call fails, an exception is thrown.

Setting the time-out value for operations

After making a function call, Web Services waits for a certain period of time for the response. If Web Services does not receive a response by the end of this period, the operation times-out. For clients using the .NET framework, the default value of this time-out period is 100000 milliseconds; the default value might vary for other development environments.

For lengthy operations, such as provisioning sites, which might take longer than the default value, you can specify a time-out value after instantiating the Web Service reference.

To specify an indefinite time-out value, set the **Timeout** property to -1.

The following is an example of the C# code used to set the Timeout property.

```
HostingWS w = new HostingWS();
w.Timeout = -1;
```

Service provider related actions

This topic lists the various service provider related operations, available in the Ensim Pro for Windows Web Service.

Viewing information of a reseller

This method displays the details of a reseller on the Ensim Pro server.

Method	public ResellerForUpdate GetResellerInfo (string resellershortname, string username, string password)
Parameters	• resellershortname, which is the short name of reseller • username, which is the username of the service provider • password, which is the password of the service provider

Return Type	Object of ResellerForUpdate with details of the specified reseller.

Example Code:

```
ResellerForUpdate resInfo = w.GetResellerInfo("myreseller.com", "admin",
"ensim123");

Console.WriteLine("DisplayName: " + resInfo.DisplayName);
Console.WriteLine("ShortName: " + resInfo.ShortName);
Console.WriteLine("NumberOfSitesCreated: " + resInfo.NumberOfSitesCreated);
//Quota values are returned as a string of values (sold, limit, usage)
Console.WriteLine("Bandwidth: " + resInfo.Bandwidth);
Console.WriteLine("DiskSpace: " + resInfo.DiskSpace);
Console.WriteLine("NumberOfUsers: " + resInfo.NumberOfUsers);
Console.WriteLine("NumberOfDNSZones: " + resInfo.NumberOfDNSZones);
Console.WriteLine("InstancesOfDNS: " + resInfo.InstancesOfDNS);
Console.WriteLine("NumberOfSubDomains: " + resInfo.NumberOfSubDomains);
Console.WriteLine("NumberOfInstallationsOfW3Svc: " +
resInfo.NumberOfInstallationsOfW3Svc);
Console.WriteLine("NumberOfInstallationsOfMSFTPSvc: " +
resInfo.NumberOfInstallationsOfMSFTPSvc);
Console.WriteLine("NumberOfInstallationsOfPHP: " +
resInfo.NumberOfInstallationsOfPHP);
Console.WriteLine("NumberOfInstallationsOfFrontPage: " +
resInfo.NumberOfInstallationsOfFrontPage);
Console.WriteLine("NumberOfInstallationsOfODBC: " +
resInfo.NumberOfInstallationsOfODBC);
Console.WriteLine("NumberOfSQLDSNs: " + resInfo.NumberOfSQLDSNs);
Console.WriteLine("NumberOfMyODBCDSNs: " + resInfo.NumberOfMyODBCDSNs);
Console.WriteLine("NumberOfAccessDSNs: " + resInfo.NumberOfAccessDSNs);
Console.WriteLine("NumberOfInstallationsOfColdfusion: " +
resInfo.NumberOfInstallationsOfColdfusion);
Console.WriteLine("InstancesOfDNS: " + resInfo.InstancesOfDNS);
Console.WriteLine("NumberOfInstallationsOfFileManager: " +
resInfo.NumberOfInstallationsOfFileManager);
Console.WriteLine("NumberOfInstallationsOfWebalizer: " +
resInfo.NumberOfInstallationsOfWebalizer);
Console.WriteLine("NumberOfInstallationsOfPerl: " +
resInfo.NumberOfInstallationsOfPerl);
Console.WriteLine("NumberOfInstallationsOfMSSQL: " +
resInfo.NumberOfInstallationsOfMSSQL);
Console.WriteLine("NumberOfMSSQLDatabases: " + resInfo.NumberOfMSSQLDatabases);
Console.WriteLine("TotalDatabasesizeOfMSSQLDatabases: " +
resInfo.TotalDatabasesizeOfMSSQLDatabases);
Console.WriteLine("NumberOfInstallationsOfPowerTools: " +
resInfo.NumberOfInstallationsOfPowerTools);
Console.WriteLine("NumberOfInstallationsOfColdfusion: " +
resInfo.NumberOfInstallationsOfColdfusion);
Console.WriteLine("NumberOfInstallationsOfMail: " +
resInfo.NumberOfInstallationsOfMail);
Console.WriteLine("NumberOfInstallationsOfMySQL: " +
resInfo.NumberOfInstallationsOfMySQL);
Console.WriteLine("NumberOfMySQLDatabases: " + resInfo.NumberOfMySQLDatabases);
Console.WriteLine("NumberOfInstallationsOfUrchin: " +
resInfo.NumberOfInstallationsOfUrchin);
```

Adding a reseller

This method creates a new reseller from an existing template on the Ensim Pro server.

Method	public void AddReseller (Reseller reseller, string username, string password)
Parameters	• reseller, which is the Reseller object with details of reseller to be added • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	none

Note: Before running this method, the Ensim Pro control panel must contain a template for adding a new reseller. If a template does not exist, you must create a new template using the Ensim Pro control panel. For instructions on creating a new template, please refer to the online Help available in the Ensim Pro control panel by clicking General Help in the *Help* section on the left navigation panel.

Example Code:

```
Reseller res = new Reseller();
res.DisplayName = "myreseller.com";
//Reseller name has to be prefixed with "reseller."
res.Name = "reseller." + "myreseller.com";
res.ShortName = "myreseller.com";
res.TemplateName = "Gold Reseller Plan";
res.ContactPhone = "0123456789";
res.ContactEmail = "blackhole@ensim.com";
res.FullName = "Test Reseller";
res.UserName = "admin";
res.Password = "ensim123";
res.Email = "user@example1.com";

w.AddReseller(res, "admin", "ensim123");
```

Updating reseller information

This method updates an existing reseller's information on the Ensim Pro server.

Method	public void UpdateReseller (ResellerForUpdate reseller, string username, string password)
Parameters	• reseller, which is the object of ResellerForUpdate class with details of reseller to be updated • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	none

Example Code:

```
ResellerForUpdate resellerInfo;
resellerInfo = w.GetResellerInfo("myreseller.com", "admin", "ensim123");

//Set general info
resellerInfo.DisplayName = "myreseller.com";
resellerInfo.ContactName = "myreseller.com";
resellerInfo.ShortName = "myreseller.com";
resellerInfo.ContactEmail = "blackhole@ensim.com";
resellerInfo.Add = 1; //1- Increase quota, 0- Decrease Quota, 2- Set quota to
specified value

//set quota values

//if Add = 0 or 1 and if quota is specified as -1 then Unlimited quota is added or
removed

// if Add = 2 and if quota is specified as -1 then unlimited quota is added.
// if Add = 2 and quota is specified as 0 then quota is removed,
//if Add = 2 and quota is other than -1 and 0 then the specified value is the
quota value
resellerInfo.Bandwidth = "10240";
resellerInfo.DiskSpace = "250";
resellerInfo.NumberOfUsers = "50";
resellerInfo.NumberOfDNSZones = "50";
resellerInfo.InstancesOfDNS = "10";
resellerInfo.NumberOfSubDomains = "100";
resellerInfo.NumberOfInstallationsOfW3Svc = "10";
resellerInfo.NumberOfInstallationsOfMSFTPSvc = "10";
resellerInfo.NumberOfInstallationsOfPHP = "10";
resellerInfo.NumberOfInstallationsOfFrontPage = "10";
resellerInfo.NumberOfInstallationsOfFileManager = "10";
resellerInfo.NumberOfInstallationsOfWebalizer = "10";
resellerInfo.NumberOfInstallationsOfPerl = "10";
resellerInfo.NumberOfInstallationsOfPowerTools = "10";
resellerInfo.NumberOfInstallationsOfColdfusion = "10";
resellerInfo.NumberOfInstallationsOfUrchin = "10";
resellerInfo.NumberOfInstallationsOfMail = "10";
resellerInfo.NumberOfInstallationsOfMSSQL = "10";
resellerInfo.NumberOfMSSQLDatabases = "10";
resellerInfo.TotalDatabasesizeOfMSSQLDatabases = "200";
resellerInfo.NumberOfInstallationsOfODBC = "10";
resellerInfo.NumberOfMyODBCDSNs = "50";
resellerInfo.NumberOfSQLDSNs = "50";
resellerInfo.NumberOfAccessDSNs = "50";
resellerInfo.NumberOfInstallationsOfMySQL = "10";
resellerInfo.NumberOfMySQLDatabases = "10";

//Call update method
w.UpdateReseller(resellerInfo, "admin", "ensim123");
```

Deleting a reseller

This method deletes an existing reseller from the Ensim Pro server.

Method	public void DeleteReseller (string resellershortname, string username, string password)

Parameters	• resellershortname, which is the short name of reseller to be deleted • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	none

Example Code:

```
w.DeleteReseller("myreseller.com", "admin", "ensim123");
```

Viewing database tools list

This method displays the existing database tools list on the Ensim Pro server.

Method	public DatabaseToolData[] GetDatabaseToolsList (DatabaseToolType databasetooltype, string username, string password)
Parameters	• databasetooltype, specifes type of database tools • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	Array of DatabaseToolData objects with details of database tools of the specified database tool type.

Example Code:

```
DatabaseToolData[] dbToolData;

//for MsSql based tools
dbToolData = w.GetDatabaseToolsList(DatabaseToolType.MSSQL2005, "admin",
"ensim123");
foreach (DatabaseToolData tool in dbToolData)
{
   System.Console.WriteLine("Tool ID: " + tool.ID);
   System.Console.WriteLine("Tool Name: " + tool.ToolName);
   System.Console.WriteLine("Tool Type: " + tool.ToolType.ToString());
   System.Console.WriteLine("Tool Target: " + tool.Target);
   System.Console.WriteLine("Tool help Target: " + tool.HelpTarget);
}

//for MySql based tools
dbToolData = w.GetDatabaseToolsList(DatabaseToolType.MySQL, "admin", "ensim123");
foreach (DatabaseToolData tool in dbToolData)
{
   System.Console.WriteLine("Tool ID: " + tool.ID);
   System.Console.WriteLine("Tool Name: " + tool.ToolName);
   System.Console.WriteLine("Tool Type: " + tool.ToolType.ToString());
   System.Console.WriteLine("Tool Target: " + tool.Target);
   System.Console.WriteLine("Tool help Target: " + tool.HelpTarget);
}

//for All types of DB tools
dbToolData = w.GetDatabaseToolsList(DatabaseToolType.All, "admin", "ensim123");
```

```
foreach (DatabaseToolData tool in dbToolData)
{
   System.Console.WriteLine("Tool ID: " + tool.ID);
   System.Console.WriteLine("Tool Name: " + tool.ToolName);
   System.Console.WriteLine("Tool Type: " + tool.ToolType.ToString());
   System.Console.WriteLine("Tool Target: " + tool.Target);
   System.Console.WriteLine("Tool help Target: " + tool.HelpTarget);
}
```

Adding a database tool

This method creates a database tool on the Ensim Pro server.

Method	public void AddDatabaseTool (DatabaseToolData databasetool, string username, string password)
Parameters	• databasetool, which is the object of DatabaseToolData to be added with all details • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	none

Note: You **cannot** use phpMyAdmin as a tool name as this is the default tool provided by Ensim Pro.

Example code:

```
DatabaseToolData dbToolData = new DatabaseToolData();
dbToolData.ToolName = "MyDatabaseTool";
dbToolData.ToolType = DatabaseToolType.MSSQL2005;
dbToolData.Target = "http://dbmgmt.myisp.net";
dbToolData.HelpTarget = "http://dbmgmthelp.myisp.net";

w.AddDatabaseTool(dbToolData, "admin", "ensim123");
```

Deleting a database tool

This method deletes a database tool from the Ensim Pro server.

Method	public void DeleteDatabaseTool (string databasetoolname, string username, string password)
Parameters	• databasetoolname, which is the name of databasetool to be deleted • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	none

Example code:

```
w.DeleteDatabaseTool("MyDatabaseTool", "admin", "ensim123");
```

Updating database tool information

This method updates an existing database tool on the Ensim Pro server.

Method	public void EditDatabaseTool (DatabaseToolData databasetool, string username, string password)
Parameters	• databasetool, which is the object of DatabaseToolData with details of the tool to be edited • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	none

Example Code:

```
DatabaseToolData dbToolData = new DatabaseToolData();
dbToolData.ToolName = "MyDatabaseTool"; // Name of existing database tool
dbToolData.ToolType = DatabaseToolType.MySQL;
dbToolData.Target = "http://dbmgmt.myisp.net";
dbToolData.HelpTarget = "http://dbmgmthelp.myisp.net";

w.EditDatabaseTool(dbToolData, "admin", "ensim123");
```

Note: The name of the database tool cannot be edited.

Updating the license

This method updates the license on the Ensim Pro server.

Method	public string UpdateLicense (string licensekey, string username, string password)
Parameters	• licensekey, which is the new license key for the product • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	none

Example code:

```
w.UpdateLicense("XXXX-XXXX-XXXX-XXXX-XXXX", "admin", "ensim123");
```

Setting the encoding type for email messages

This method is used for setting the encoding type for email messages, on the Ensim Pro server.

Method	public void SetEncodingType (string EncodingType, string username, string password)
Parameters	• EncodingType, which is the encoding type to be used for email messages. • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	none

Example code:

```
w.SetEncodingType("utf-8", "admin", "ensim123");
```

Setting the percent quota usage threshold

This method is used for setting the percent threshold for quota usage on the Ensim Pro server. Quota usage emails will be generated when the quota usage exceeds the specified value.

Method	public void SetQuotaUsageThreshold(int QuotaUsageThreshold, string username, string password)
Parameters	• QuotaUsageThreshold, which is the percent threshold to be used for quota usage. • username, which is the username of the service provider • password, which is the password of the service provider
Return Value	none

Example code:

```
w.SetQuotaUsageThreshold (85, "admin", "ensim123");
```

Service provider/Reseller related actions

This topic lists the various the service provider/Reseller related operations, available in the Ensim Pro for Windows Web Service.

Viewing information of a site

This method displays the details of a site on the Ensim Pro server.

Method	public SiteInformation GetSiteInfo (string siteshortname, string username, string password)
Parameters	• siteshortname, which is the short name of the site • username, which is the username of the service provider/Reseller • password, which is the password of the service provider/Reseller
Return Value	Object of SiteInformation containing details of the specified site.

Example code:

```
SiteInformation siteInfo;
siteInfo = w.GetSiteInfo("mysite.com", "admin", "ensim123");

Console.WriteLine("Site Display Name: " + siteInfo.DisplayName);
Console.WriteLine("Domain Name: " + siteInfo.DomainName);
Console.WriteLine("Short Name: " + siteInfo.ShortName);
Console.WriteLine("Admin User Name: " + siteInfo.AdminUserName);
Console.WriteLine("Contact Email: " + siteInfo.ContactEmail);
Console.WriteLine("Contact Name: " + siteInfo.ContactName);
Console.WriteLine("Contact Phone: " + siteInfo.ContactPhone);
```

Adding a site

This method creates a new site from an existing template on the Ensim Pro server.

Method	public void AddSite (Site site, string username, string password)
Parameters	• site, which is the Site object with details of the new site to be added • username, which is the username of the service provider/Reseller • password, which is the password of the service provider/Reseller
Return Value	none

Note: Before running this method, the Ensim Pro control panel must contain a template for adding a new site. If a template does not exist, you must create a new template using the Ensim Pro control panel. For instructions on creating a new template, please refer to the online Help available in the Ensim Pro control panel by clicking General Help in the *Help* section on the left navigation panel.

Example Code:

```
Site site = new Site();
site.DisplayName = "mysite.com";
site.Name = "mysite.com";
site.ShortName = "mysite.com";
site.TemplateName = "Default Template";
```

```
site.ContactEmail = "blackhole@ensim.com";
site.AdminUserName = "admin";
site.AdminDisplayName = "Administrator";
site.AdminPassword = "ensim123";
site.AdminEmail = "user@example1.com";
site.IpBased = true; //Set false for name based sites
site.IpSelection = "auto";
//Specify IP address(10.76.32.95) or auto for automatic allocation.
//(This is applicable only for IP-based sites)
w.AddSite(site, "admin", "ensim123");
```

Updating site information

This method updates information of the site on the Ensim Pro server.

Method	public void UpdateSite (SiteForUpdate site, string username, string password)
Parameters	• site, which is the object of SiteForUpdate with all required details of the site to be updated • username, which is the username of the service provider/Reseller • password, which is the password of the service provider/Reseller
Return Value	none

Example Code:

```
SiteForUpdate site = new SiteForUpdate();
WebHosting webHostingConfig = new WebHosting();

//General details
site.DisplayName = "mysite.com";
site.ContactName = "ABC";
site.ShortName = "mysite.com";
site.ContactName = "0132465798";
site.ContactEmail = "blackhole@ensim.com";
site.AdminDisplayName = "Administrator";
site.AdminUserName = "admin@mysite.com";
site.AdminPassword = "ensim123";
site.AdminEmail = "user@example1.com";

//Service Details
webHostingConfig.Enabled = true;
webHostingConfig.IsIpbased = true;
webHostingConfig.BandwidthQuota = 1024;
webHostingConfig.BillingCycleStartDay = 1;
webHostingConfig.BandwidthActionSendMail = true;
webHostingConfig.BandwidthCapped = false;
webHostingConfig.DiskQuota = 25;
webHostingConfig.MaxUsers = 5;

//Updating Service Component: msftp
msftpsvc ftpConfig = new msftpsvc();
ftpConfig.Enabled = true;
//true- Component is added or updated, false- component is removed
ftpConfig.UserIsolationMode = 0;
ftpConfig.AnonymousConnections = true;
```

```
ftpConfig.MaxConnections = 100;
ftpConfig.Timeout = 9999;
ftpConfig.SiteLogs = true;
ftpConfig.LogType = enumFTPLogType.W3CExtendedLogFileFormat;
ftpConfig.IISPermRead = true;
ftpConfig.IISPermWrite = false;
ftpConfig.MessageWelcome = "Welcome";
ftpConfig.MessageExit = "Thank You";
ftpConfig.MessageMaxClients = "Maximum number of Clients are already connected.
Try Later.";

//Call update method
site.webhosting = webHostingConfig;
site.webhosting._msftpsvc = ftpConfig;
w.UpdateSite(site, "admin", "ensim123");
```

Note: While updating any site configuration, site general details and Web hosting configuration details must specified.

Deleting a site

This method deletes an existing site on the Ensim Pro server.

Method	public void DeleteSite (string siteshortname, string username, string password)
Parameters	• siteshortname, which is the short name of site to be deleted • username, which is the username of the service provider/Reseller • password, which is the password of the service provider/Reseller
Return Value	none

Example Code:

```
w.DeleteSite("mysite.com", "admin", "ensim123");
```

Importing a site

This method imports the site's configuration.

Method	public void ImportSite (SiteExportImport[] exportimportoptions, string username, string password)
Parameters	• exportimportoptions, which is the Array of SiteExportImport objects with details of all sites to be imported • username, which is the username of the service provider/Reseller • password, which is the password of the service provider/Reseller
Return Value	none

Example Code:

```
SiteExportImport[] eximOptions = new SiteExportImport[] {new SiteExportImport()};

eximOptions[0].SiteName = "mysite.com";
eximOptions[0].SiteShortName = "mysite.com";
eximOptions[0].ConfigOnly = false;
//Complete path to site's exported data folder has to be specified
eximOptions[0].ExportImportPath =
"C:\\Export\\ISP\\EPWCHK\\Site\\wstestorg.raj\\12-04-06 18-25-02 0";
eximOptions[0].WithAllUsers = true;

w.ImportSite(eximOptions, "admin", "ensim123");
```

Exporting a site

This method exports the site's configuration and /or users to a folder structure.

Method	public void ExportSite (SiteExportImport[] exportimportoptions, string username, string password)
Parameters	• exportimportoptions, which is the array of SiteExportImport objects with details of all sites to be exported • username, which is the username of the service provider/Reseller • password, which is the password of the service provider/Reseller
Return Value	none

Example Code:

```
SiteExportImport[] eximOptions = new SiteExportImport[] {new SiteExportImport()};

eximOptions[0].SiteName = "mysite.com";
eximOptions[0].SiteShortName = "mysite.com";
eximOptions[0].ConfigOnly = false;
eximOptions[0].ExportImportPath = "C:\\Export";
eximOptions[0].WithAllUsers = true;

w.ExportSite(eximOptions, "admin", "ensim123");
```

Site administrator related actions

Adding a user

This method creates a new user on the Ensim Pro server.

Method	public void AddUser (User user, string username, string password)

Parameters	• user, which is the User object with details of user to be created • username, which is the username of the site administrator • password, which is the password of the site administrator
Return Value	none

Example Code:

```
User user = new User();

//Mandatory parameters
user.Username = "user1";
user.Password = "ensim123";
user.Template = "FirstUserTemplate";

//Optional parameters
user.Enabled = true;
user.Admin = false;
user.City = "Sunnyvale";
user.Company = "Ensim";
user.Country = "US";
user.Department = "Engg.";
user.Description = "WS Test User";
user.Email = "blackhole@ensim.com";
user.FirstName = "WebServices";
user.LastName = "Test User";
user.MiddleName = "";
user.Office = "Office";
user.Phone = "0123465789";
user.State = "CA";
user.Street = "Office Lane";

w.AddUser(user, "admin@mysite.com", "ensim123");
```

NOTE:Before running this method, the Ensim Pro control panel must contain a template for adding a new user for that site. If a template does not exist, you must create a new template using the Ensim Pro control panel or by using Web Services API (page 25).

Updating user information

This method updates an existing user's information.

Method	public void UpdateUser (UserForUpdate user, string username, string password)
Parameters	• user, which is the UserForUpdate object with details for the user to be updated • username, which is the username of the site administrator • password, which is the password of the site administrator
Return Value	none

Example Code:

```
UserForUpdate user = new UserForUpdate();

user.Username = "user1@mysite.com"; //username cannot be changed
user.Enabled = true;
user.Admin = false;
user.City = "Sunnyvale";
user.Company = "Ensim";
user.Country = "US";
user.Department = "Engg.";
user.Description = "WS Test User";
user.Email = "blackhole@ensim.com";
user.FirstName = "WebServices";
user.LastName = "Test User";
user.MiddleName = "";
user.Office = "Office";
user.Phone = "0123465789";
user.State = "CA";
user.Street = "Office Lane";

//user service components
user.mail = null;       //To remove service component set to null
user.msftpsvc = null; //To remove service component set to null
user.w3svc = new User_W3SVC();
user.frontpage = new User_FRONTPAGE();
user.frontpage.ContentUploader = true; //Allow user access to site's website

w.UpdateUser(user, "admin@mysite.com", "ensim123");
```

Note: While updating the user information, the user name cannot be changed

Deleting a user

This method deletes an existing user from the Ensim Pro server.

Method	public void DeleteUser (string name, string username, string password)
Parameters	• name, which is the complete username with domain name of the user to be deleted • username, which is the username of the site administrator • password, which is the password of the site administrator
Return Value	none

Example Code:

```
w.DeleteUser("user1@mysite.com", "admin@mysite.com", "ensim123");
```

Adding a user template

This method adds a user template, which can be used later for creating a new user.

Method	public void AddUserTemplate (UserTemplate userTemplate, string username, string password)
Parameters	• userTemplate, which is the UserTemplate object with details of the new user template to be added. • username, which is the username of the site administrator • password, which is the password of the site administrator
Return Value	none

Example Code:

```
UserTemplate userTemplate = new UserTemplate();

userTemplate.Name = "FirstUserTemplate";
userTemplate.Description = "FirstUserTemplate";
userTemplate.Enabled = true;
userTemplate.w3svc = new User_W3SVC();
userTemplate.msftpsvc = new User_MSFTPSVC();
userTemplate.msftpsvc.ContentUploader = true;
userTemplate.msftpsvc.IISPermRead = true;
userTemplate.msftpsvc.IISPermWrite = false;
userTemplate.frontpage = new User_FRONTPAGE();
userTemplate.frontpage.ContentUploader = true;
userTemplate.mail = new User_MAIL();
userTemplate.mail.UserEnableAutoresponders = true;
userTemplate.mail.UserEnableForwards = true;
userTemplate.mail.UserEnableWebAccess = true;

w.AddUserTemplate( userTemplate, "admin@mysite.com", "ensim123");
```

Deleting a user template

This method deletes an existing user template from the Ensim Pro server.

Method	public void DeleteUserTemplate (string userTemplateName, string username, string password)
Parameters	• userTemplateName, which is the name of user template to be deleted. • username, which is the username of the site administrator • password, which is the password of the site administrator
Return Value	none

Example Code:

```
w.DeleteUserTemplate("FirstUserTemplate", "admin@mysite.com", "ensim123");
```

Adding a Microsoft SQL Server 2005 database

This method adds a Microsoft SQL Server 2005 database, to the Ensim Pro server.

Method	public void AddMSSQLDatabase (string DatabaseName, string username, string password)
Parameters	• DatabaseName, which is the name of the SQL 2005 database to be created • username, which is the username of the site administrator • password, which is the password of the site administrator
Return Value	none

Note: For successful Microsoft SQL Server 2005 database creation, the server administrator must select the option **enable database management** during site creation.

Example Code:

```
w.AddMSSQLDatabase("wstestdb", "admin@mysite.com", "ensim123");
```

Deleting a Microsoft SQL Server 2005 database

This method deletes an existing Microsoft SQL Server 2005 database, from the Ensim Pro server.

Method	public void DeleteMSSQLDatabase (string DatabaseName, string username, string password)
Parameters	• DatabaseName, which is the name of the SQL 2005 database to be deleted. • username, which is the username of the site administrator • password, which is the password of the site administrator
Return Value	none

Example Code:

```
//To delete database specify complete name of database
w.DeleteMSSQLDatabase("mysite_com_db___wstestdb", "admin@mysite.com", "ensim123");
```

APPENDIX A

Detailed sample codes

This section provides detailed sample codes, which can be used for manipulating individual service components and their parameters for sites.

IIS Web Service

This code sample explains how to add/update or remove IIS Web Service.

```
SiteForUpdate site = new SiteForUpdate();

site.webhosting = new WebHosting();

//Add IIS Web

    w3svc w3svcConfig = new w3svc();
    w3svcConfig.Enabled = true;             // To remove set to false
    w3svcConfig.EnableDotNet = true;
    w3svcConfig.ASPNetVersion = "2.0.50727.0";
    w3svcConfig.DefaultSSL = true;
    w3svcConfig.SSL = true;
    w3svcConfig.SharedSSL = false;
    w3svcConfig.ThirdyPartySSLCertificateInstalled = false;
    w3svcConfig.SSI = true;
    w3svcConfig.SiteLogs = true;
    w3svcConfig.HostHeaders = true;
    w3svcConfig.UseAsPDH = true;
    w3svcConfig.MaxSubDomains = 100;
    w3svcConfig.LogType = enumW3SVCLogType.W3CExtendedLogFileFormat;
    w3svcConfig.ChangeLogFormat = true;
    w3svcConfig.Timeout = 9999;
    w3svcConfig.MaxConnections = 10000;
    w3svcConfig.Hits = 2;
    w3svcConfig.EnableBWQuota = true;
    w3svcConfig.BandwidthQuota = 2048;
    w3svcConfig.IISPermRead = true;
    w3svcConfig.IISPermWrite = true;
```

```
    w3svcConfig.IISPermDirectoryBrowsing = true;
    w3svcConfig.IISPermExec = 2;
    w3svcConfig.IISApplProtection = 1;      // 1- Dedicated pool, 0 - shared pool
    site.webhosting._w3svc = new w3svc();
    site.webhosting._w3svc = w3svcConfig;
//General details
    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";
//Service Details
    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;
//Call update method
    w.UpdateSite(site, "admin", "ensim123");
```

PHP

This code sample explains how to add/update or remove PHP.

```
SiteForUpdate site = new SiteForUpdate();
site.webhosting = new WebHosting();
//Add PHP
    php phpConfig = new php();
    phpConfig.Enabled = true;  // To remove set to false
    phpConfig.PhpType = 1;

    site.webhosting._php = new php();
```

```
    site.webhosting._php = phpConfig;
//General details
    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";
//Service Details
    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;
//Call update method
    w.UpdateSite(site, "admin", "ensim123");
```

FrontPage

This code sample explains how to add/update or remove FrontPage.

```
SiteForUpdate site = new SiteForUpdate();
site.webhosting = new WebHosting();
//Add frontpage
    frontpage fpConfig = new frontpage();
    fpConfig.Enabled = true;   // To remove set to false
    fpConfig.EnableAuthoring = true;
    fpConfig.RequireSSL = true;
    fpConfig.MailReplyTo = "blackhole@ensim.com";
    fpConfig.MailSender = "webmaster@mysite.com";
    fpConfig.SMTPHost = "smtp.myisp.net";
    site.webhosting._frontpage = new frontpage();
    site.webhosting._frontpage = fpConfig;
```

```
//General details
    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";
//Service Details
    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;
//Call update method
    w.UpdateSite(site, "admin", "ensim123");
```

ODBC

This code sample explains how to add/update or remove ODBC.

```
SiteForUpdate site = new SiteForUpdate();
site.webhosting = new WebHosting();
//Add ODBC
    odbc odbcConfig = new odbc();
    odbcConfig.Enabled = true; // To remove set to false
    odbcConfig.TotalAccessDSNCount = 5;
    odbcConfig.TotalMyODBCDSNCount = 5;
    odbcConfig.TotalSQLDSNCount = 5;
    site.webhosting._odbc = new odbc();
    site.webhosting._odbc = odbcConfig;
//General details
    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
```

```
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";
//Service Details
    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;
//Call update method
    w.UpdateSite(site, "admin", "ensim123");
```

File Manager

This code sample explains how to add/update or remove File Manager.

```
SiteForUpdate site = new SiteForUpdate();
site.webhosting = new WebHosting();
//Add file manager
    filemanager fmConfig = new filemanager();
    fmConfig.Enabled = true;          // To remove set to false
    site.webhosting._filemanager = new filemanager();
    site.webhosting._filemanager = fmConfig;
//General details
    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
```

```
    site.AdminEmail = "user@example1.com";
//Service Details
    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;
//Call update method
    w.UpdateSite(site, "admin", "ensim123");
```

Webalizer

This code sample explains how to add/update or remove Webalizer.

```
SiteForUpdate site = new SiteForUpdate();
site.webhosting = new WebHosting();
//Add Webalizer
    webalizer webConfig = new webalizer();
    webConfig.Enabled = true;  // To remove set to false

    site.webhosting._webalizer = new webalizer();
    site.webhosting._webalizer = webConfig;
//General details
    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";
//Service Details
    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
```

```
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;
//Call update method
    w.UpdateSite(site, "admin", "ensim123");
```

Active Perl

This code sample explains how to add/update or remove Active Perl.

```
SiteForUpdate site = new SiteForUpdate();
site.webhosting = new WebHosting();
//add Perl
    perl perlConfig = new perl();
    perlConfig.Enabled = true; // To remove set to false
    site.webhosting._perl = new perl();
    site.webhosting._perl = perlConfig;
//General details
site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";
//Service Details
    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;
//Call update method
```

```
    w.UpdateSite(site, "admin", "ensim123");
```

SQL 2005

This code sample explains how to add/update or remove SQL 2005.

```
SiteForUpdate site = new SiteForUpdate();

site.webhosting = new WebHosting();

//add SQL 2005

    mssql mssqlConfig = new mssql();
    mssqlConfig.Enabled = true;      // To remove set to false
    mssqlConfig.TotalDatabaseCount = 1;
    mssqlConfig.TotalDatabaseSize = 20;
    mssqlConfig.EnableDBMgmt = true;
    mssqlConfig.DatabaseTools = "1";
    site.webhosting._mssql = new mssql();
    site.webhosting._mssql = mssqlConfig;

//General details

    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";

//Service Details

    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;

//Call update method

    w.UpdateSite(site, "admin", "ensim123");
```

MySQL

This code sample explains how to add/update or remove MySQL.

```
SiteForUpdate site = new SiteForUpdate();

site.webhosting = new WebHosting();

//add MySQL

    mysql mysqlConfig = new mysql();
    mysqlConfig.Enabled = true;      // To remove set to false
    mysqlConfig.TotalDatabaseCount = 1;
    mysqlConfig.DatabaseTools = "2";
    site.webhosting._mysql = new mysql();
    site.webhosting._mysql = mysqlConfig;

//General details

    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";

//Service Details

    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;

//Call update method

    w.UpdateSite(site, "admin", "ensim123");
```

ColdFusion

This code sample explains how to add/update or remove ColdFusion.

```
SiteForUpdate site = new SiteForUpdate();

site.webhosting = new WebHosting();

//add Coldfusion

   coldfusion cfConfig = new coldfusion();
   cfConfig.Enabled = true;          // To remove set to false
   site.webhosting._coldfusion = new coldfusion();
   site.webhosting._coldfusion = cfConfig;

//General details

   site.DisplayName = "mysite.com";
   site.ContactName = "ABC";
   site.ShortName = "mysite.com";
   site.ContactName = "0132465798";
   site.ContactEmail = "blackhole@ensim.com";
   site.AdminDisplayName = "Administrator";
   site.AdminUserName = "admin@mysite.com";
   site.AdminPassword = "ensim123";
   site.AdminEmail = "user@example1.com";

//Service Details

   site.webhosting.Enabled = true;
   site.webhosting.IsIpbased = true;
   site.webhosting.BandwidthQuota = 1024;
   site.webhosting.BillingCycleStartDay = 1;
   site.webhosting.BandwidthActionSendMail = true;
   site.webhosting.BandwidthCapped = false;
   site.webhosting.DiskQuota = 25;
   site.webhosting.MaxUsers = 5;

//Call update method

   w.UpdateSite(site, "admin", "ensim123");
```

PowerTools

This code sample explains how to add/update or remove Power Tools.

```
SiteForUpdate site = new SiteForUpdate();

site.webhosting = new WebHosting();

//add Powertools

   powertools ptoolConfig = new powertools();
   ptoolConfig.Enabled = true;            // To remove set to false
```

```
    ptoolConfig.PowerToolList = "1,2,3,4,5";
    // For Fresh Install Setups
    // 1 represents Gallery
    // 2 represents WordPress Web log
    // 3 represents Community server
    // 4 represents DotNetNuke
    // 5 represents PhpBB bulletin board

    // For Upgraded Setups
    // 1 represents Gallery
    // 2 represents WordPress Web log
    // 3 represents Community server 1.1
    // 4 represents DotNetNuke 3.2.0
    // 5 represents PhpBB bulletin board
    // 6 represents Community server 2.1
    // 7 represents DotNetNuke 4.4.1

    site.webhosting._powertools = new powertools();
    site.webhosting._powertools = ptoolConfig;
//General details
    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";
//Service Details
    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;
//Call update method
    w.UpdateSite(site, "admin", "ensim123");
```

Urchin

This code sample explains how to add/update or remove Urchin.

```
SiteForUpdate site = new SiteForUpdate();

site.webhosting = new WebHosting();

//add Urchin

    urchin urchinConfig = new urchin();
    urchinConfig.Enabled = true;            // To remove set to false
    site.webhosting._urchin = new urchin();
    site.webhosting._urchin = urchinConfig;

//General details

    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";

//Service Details

    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;

//Call update method

    w.UpdateSite(site, "admin", "ensim123");
```

FTP

This code sample explains how to add/update or remove FTP.

```
SiteForUpdate site = new SiteForUpdate();

site.webhosting = new WebHosting();
```

```
//add msftp
    msftpsvc ftpConfig = new msftpsvc();
    ftpConfig.Enabled = true;  // To remove set to false
    ftpConfig.UserIsolationMode = 0;
    ftpConfig.AnonymousConnections = true;
    ftpConfig.MaxConnections = 100;
    ftpConfig.Timeout = 9999;
    ftpConfig.SiteLogs = true;
    ftpConfig.LogType = enumFTPLogType.MicrosoftIISLogFileFormat;
    ftpConfig.IISPermRead = true;
    ftpConfig.IISPermWrite = false;
    ftpConfig.MessageWelcome = "Welcome message";
    ftpConfig.MessageExit = "Exit Message";
    ftpConfig.MessageMaxClients = "Max Client Message";
    site.webhosting._msftpsvc = new msftpsvc();
    site.webhosting._msftpsvc = ftpConfig;
//General details
    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";
//Service Details
    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;
//Call update method
    w.UpdateSite(site, "admin", "ensim123");
```

DNS

This code sample explains how to add/update or remove DNS.

```
SiteForUpdate site = new SiteForUpdate();

site.webhosting = new WebHosting();

//add msdns

    msdns dnsConfig = new msdns();
    dnsConfig.Enabled = true;  // To remove set to false
    dnsConfig.NumDNSZones = 5;
    dnsConfig.AllowSubdomainsOnly = true;
    dnsConfig.AllowReverseZone = true;
    site.webhosting._msdns = new msdns();
    site.webhosting._msdns = dnsConfig;

//General details

    site.DisplayName = "mysite.com";
    site.ContactName = "ABC";
    site.ShortName = "mysite.com";
    site.ContactName = "0132465798";
    site.ContactEmail = "blackhole@ensim.com";
    site.AdminDisplayName = "Administrator";
    site.AdminUserName = "admin@mysite.com";
    site.AdminPassword = "ensim123";
    site.AdminEmail = "user@example1.com";

//Service Details

    site.webhosting.Enabled = true;
    site.webhosting.IsIpbased = true;
    site.webhosting.BandwidthQuota = 1024;
    site.webhosting.BillingCycleStartDay = 1;
    site.webhosting.BandwidthActionSendMail = true;
    site.webhosting.BandwidthCapped = false;
    site.webhosting.DiskQuota = 25;
    site.webhosting.MaxUsers = 5;

//Call update method

    w.UpdateSite(site, "admin", "ensim123");
```

ENSIM PRO - WINDOWS

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