

Parallels[®] Pro Control Panel

Parallels Pro Control Panel 10.3.1 for Windows 2003 Web Services API Guide

Legal and Copyright Notice

ISBN: N/A

Parallels

660 SW 39th Street

Suite 205

Renton, Washington 98057

USA

Phone: +1 (425) 282 6400

Fax: +1 (425) 282 6444

© Copyright 2007 Ensim Corporation. All rights reserved.

© Copyright 2008 Parallels, Inc. All rights reserved.

Distribution of this work or derivative of this work in any form is prohibited unless prior written permission is obtained from the copyright holder.

Product and service names mentioned herein are the trademarks of their respective owners.

Contents

Preface	5
Typographical Conventions	5
Feedback and Support	6
About This Guide	7
Intended Audience	7
Related Documentation	7
Web Services API Calls	8
URL for the Web Services	8
Simple C# Client	9
Step 1. Create a New Client Application in Visual Studio .NET	9
Step 2. Add the Web Reference to Your Project.....	10
Step 3. Add the Code to Provision a Site	11
Setting the Time-out Value for Operations	12
Service Provider Related Actions	13
Viewing Information of a Reseller.....	14
Adding a Reseller	15
Updating Reseller Information.....	16
Deleting a Reseller	17
Viewing Database Tools List.....	18
Adding a Database Tool.....	19
Deleting a Database Tool.....	20
Updating Database Tool Information	20
Updating the License.....	21
Setting the Encoding Type for Email Messages	21
Setting the Percent Quota Usage Threshold	22
Service Provider/Reseller Related Actions.....	22
Viewing Information of a Site.....	23
Adding a Site	24
Updating Site Information.....	25
Deleting a Site	26
Importing a Site	27
Exporting a Site	28
Site Administrator Related Actions	28
Adding a User.....	29
Updating User Information	29
Deleting a User.....	31
Adding a User Template.....	32
Deleting a User Template.....	33
Adding a Microsoft SQL Server 2005 Database	33
Deleting a Microsoft SQL Server 2005 Database	34
Detailed Sample Codes	35
IIS Web Service	36
PHP.....	37
FrontPage	38
ODBC	39

File Manager.....	40
Webalizer.....	41
Active Perl.....	42
SQL 2005.....	43
MySQL.....	44
ColdFusion.....	45
PowerTools.....	46
Urchin.....	47
FTP.....	48
DNS.....	49

Preface

In this section:

Typographical Conventions	5
Feedback and Support	6

Typographical Conventions

Before you start using this guide, it is important to understand the documentation conventions used in it.

The following kinds of formatting in the text identify special information.

<u>Formatting convention</u>	<u>Type of Information</u>	<u>Example</u>
Special Bold	Items you must select, such as menu options, command buttons, or items in a list.	Go to the System tab.
	Titles of chapters, sections, and subsections.	Read the Basic Administration chapter.
<i>Italics</i>	Used to emphasize the importance of a point, to introduce a term or to designate a command line placeholder, which is to be replaced with a real name or value.	The system supports the so called <i>wildcard character</i> search.
Monospace	The names of commands, files, directories, and domain names.	The license file is located in the <code>http://docs/common/licenses</code> directory.

Preformatted	On-screen computer output in your command-line sessions; source code in XML, C++, or other programming languages.	<pre># ls -al /files total 14470</pre>
Preformatted Bold	What you type, contrasted with on-screen computer output.	<pre># cd /root/rpms/php</pre>
CAPITALS	Names of keys on the keyboard.	SHIFT, CTRL, ALT
KEY+KEY	Key combinations for which the user must press and hold down one key and then press another.	CTRL+P, ALT+F4

Feedback and Support

To take advantage of Parallels Pro Control Panel (formerly known as Ensim Pro) support services or to find additional product documentation, visit Parallels Pro Control Panel Online Support at <http://www.parallels.com/en/support/pro/>.

To log in to Parallels Pro Control Panel online support, submit the form at <https://www.parallels.com/en/support/pro/form/>.

If you have found a mistake in this guide, or if you have suggestions or ideas on how to improve this guide, please send your feedback using the online form at <http://www.parallels.com/en/support/usersdoc/>. Please include in your report the guide's title, chapter and section titles, and the fragment of text in which you have found an error.

About This Guide

This document provides an overview of Web Services API for Parallels Pro for Windows 2003 (formerly known as Ensim Pro for Windows 2003).

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 Parallels Pro for Windows documentation section of the Parallels Pro Control Support page, <http://www.parallels.com/en/support/pro/>, 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 1).

In this section:

Intended Audience.....	7
Related Documentation	7

Intended Audience

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

The target audiences for Parallels Pro Control Panel 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 Parallels Pro Control Panel for Windows Web services

Related Documentation

For a list of related documentation, see the Parallels Pro Control Panel for Windows release notes. Documentation is also available in the Parallels Pro Control Panel for Windows section of the Parallels Pro Support site, <http://www.parallels.com/en/support/pro/>.

Web Services API Calls

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

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

This section describes the Web Services API calls that you can use on the Parallels Pro Control Panel 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 Parallels Pro for Windows license
- Set the encoding type for email messages

In this chapter:

URL for the Web Services	8
Simple C# Client.....	9
Setting the Time-out Value for Operations.....	12
Service Provider Related Actions	13
Service Provider/Reseller Related Actions	22
Site Administrator Related Actions	28

URL for the Web Services

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

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

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

Simple C# Client

This section explains how to create a C# client for Parallels Pro Control Panel 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 Parallels Pro Control Panel server.

➤ **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
```

In this section:

Step 1. Create a New Client Application in Visual Studio .NET.....	9
Step 2. Add the Web Reference to Your Project.....	10
Step 3. Add the Code to Provision a Site.....	11

Step 1. 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.

Step 2. 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 epwstest.provisioningservice;
```

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

Step 3. 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
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
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 1)
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

PART 1. SET UP CONNECTION INFORMATION FOR THE WEB SERVICE

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

The created site 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 Parallels Pro Control Panel for Windows Web Service.

In this section:

Viewing Information of a Reseller	14
Adding a Reseller	15
Updating Reseller Information	16
Deleting a Reseller	17
Viewing Database Tools List	18
Adding a Database Tool	19
Deleting a Database Tool	20
Updating Database Tool Information	20
Updating the License.....	21
Setting the Encoding Type for Email Messages.....	21
Setting the Percent Quota Usage Threshold	22

Viewing Information of a Reseller

This method displays the details of a reseller on the Parallels Pro Control Panel server.

Method	public ResellerForUpdate GetResellerInfo (string resellershortname, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel server.

Method	public void AddReseller (Reseller reseller, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels 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 Parallels Pro Control Panel. For instructions on creating a new template, please refer to the online Help available in the Parallels 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 Parallels Pro Control Panel server.

Method	public void UpdateReseller (ResellerForUpdate reseller, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel server.

Method	public void DeleteReseller (string resellershortname, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel server.

Method	public DatabaseToolData[] GetDatabaseToolsList (DatabaseToolType databasetooltype, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ databasetooltype, specifies 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 Parallels Pro Control Panel server.

Method	public void AddDatabaseTool (DatabaseToolData databasetool, string username, string password)
Parameters	<ul style="list-style-type: none">▪ 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 Parallels Pro Control Panel.

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 Parallels Pro Control Panel server.

Method	public void DeleteDatabaseTool (string databasetoolname, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel server.

Method	public void EditDatabaseTool (DatabaseToolData databasetool, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel server.

Method	public string UpdateLicense (string licensekey, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro server.

Method	public void SetEncodingType (string EncodingType, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel 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	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel for Windows Web Service.

In this section:

Viewing Information of a Site	23
Adding a Site	24
Updating Site Information	25
Deleting a Site	26
Importing a Site	27
Exporting a Site	28

Viewing Information of a Site

This method displays the details of a site on the Parallels Pro Control Panel server.

Method	public SiteInformation GetSiteInfo (string sitedshortname, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ sitedshortname, 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 Parallels Pro Control Panel server.

Method	public void AddSite (Site site, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels 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 Parallels Pro Control Panel. For instructions on creating a new template, please refer to the online Help available in the Parallels 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 Parallels Pro Control Panel server.

Method	public void UpdateSite (SiteForUpdate site, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel server.

Method	public void DeleteSite (string sitedshortname, string username, string password)
Parameters	<ul style="list-style-type: none">▪ sitedshortname, 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	<ul style="list-style-type: none"> ▪ 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	<ul style="list-style-type: none"> ▪ 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

In this section:

Adding a User	29
Updating User Information.....	29
Deleting a User	31
Adding a User Template.....	32
Deleting a User Template.....	33
Adding a Microsoft SQL Server 2005 Database	33
Deleting a Microsoft SQL Server 2005 Database	34

Adding a User

This method creates a new user on the Parallels Pro Control Panel server.

Method	public void AddUser (User user, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 = "Parallels Pro";
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 Parallels 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 Parallels Pro Control Panel or by using Web Services API (on page 32).

Updating User Information

This method updates an existing user's information.

Method	public void UpdateUser (UserForUpdate user, string username, string password)
---------------	---

Parameters	<ul style="list-style-type: none">▪ 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 = "Parallels Pro";
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 Parallels Pro Control Panel server.

Method	public void DeleteUser (string name, string username, string password)
Parameters	<ul style="list-style-type: none">▪ 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	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel server.

Method	public void DeleteUserTemplate (string userTemplateName, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel server.

Method	public void AddMSSQLDatabase (string DatabaseName, string username, string password)
Parameters	<ul style="list-style-type: none"> ▪ 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 Parallels Pro Control Panel server.

Method	public void DeleteMSSQLDatabase (string DatabaseName, string username, string password)
Parameters	<ul style="list-style-type: none">▪ 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");
```

Detailed Sample Codes

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

In this chapter:

IIS Web Service.....	36
PHP.....	37
FrontPage	38
ODBC.....	39
File Manager	40
Webalizer	41
Active Perl.....	42
SQL 2005.....	43
MySQL.....	44
ColdFusion.....	45
PowerTools	46
Urchin.....	47
FTP	48
DNS	49

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.ThirdPartySSLCertificateInstalled = 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");
```