



Installation and Deployment

Help Documentation

Installation and Deployment

Browser Requirements

SmarterStats is fully supported by the browsers below.

- Internet Explorer 6 and higher
- Google Chrome 2 and higher
- FireFox 2 and higher
- Safari 3 and higher
- Opera 10 and higher

SmarterStats System Requirements

SmarterStats was designed to operate efficiently with multiple applications on the same server based on the following minimum requirements:

- Windows XP, Windows Vista, or Windows 2003 or higher
- Microsoft .Net 4.0
- SmarterStats Web server included with product*

One of the unique benefits of SmarterStats is its ability to manage the resources it consumes.

Therefore, the resources used by SmarterStats can be adjusted to work in any environment with a variety of other desktop or server applications. For example: On a machine with multiple applications running, the system requirements can be adjusted to account for a number of different factors on the server (such as other processes that are running, the size of individual websites, the amount of traffic, the size of the log files generated, etc.). For more information, see [Deployment Guides](#) .

*SmarterStats includes a basic Web server, so the product is fully function upon installation--even without the existence of IIS or other Web servers. However, SmarterTools recommends installing Microsoft IIS 7.0 in place of the SmarterStats Web server for increased performance and security in medium to high-volume environments. For more information, see [Running SmarterStats as an IIS Site](#) .

SmarterStats Enterprise Remote Service

If you are running SmarterStats Enterprise with the remote service, SmarterTools recommends the following requirements:

- Windows XP, Windows Vista, or Windows 2003 or higher
- Microsoft .NET 4.0 Framework (including all applicable service packs and/or patches)

Note: IIS 7.0 requires that the IIS 6.0 Management Compatibility (and all subsets therein) to be installed.

The remote service can run on any server on which log files reside. It can also use FTP to pull the log files from other servers or it can be configured to pull log files from a network location. Most sites will require the service to use 256MB of RAM or less. However, sites with an abnormal amount of unique page names or sites with extremely large log files (greater than 1GB per day) could require more RAM. Please keep this in mind when looking at a specific Web server's hardware configuration (CPU, available memory, etc.).

In addition, the Web interface communicates with the remote service via port 50003. Firewalls or any other security measures that might prohibit communication from this port must have rules established to open this port.

Note: Every installation and environment is unique. Extra load caused by excessive volume and/or other factors may require more disk space, memory, database allocation, etc. No warrantee or guarantee is expressed or implied as to the efficacy or viability of these recommendations for a particular environment or application.

Installation

SmarterStats comes as a single installation file that contains everything necessary to run the product. The SmarterStats installer can be downloaded from www.smartertools.com. Both the free and the licensed editions of SmarterStats are contained within this installation file, so it is not necessary to download or install the file again if you purchase a license after trying the free edition.

Please refer to the KB article [How To - Install SmarterStats](#) for step-by-step installation instructions.

Note: If you plan to install SmarterStats Enterprise on a remote server, use the service MSI installer package in addition to the standard installer.

Also, Windows Server 2003 users may need to install SmarterStats from Add/Remove Programs in the control panel in order for file permissions to be properly set.

Running as an IIS Site

By default, SmarterStats installs a basic Web server that allows companies to start using the application immediately after installation. However, SmarterTools recommends moving to a more robust and secure Web server, such as Microsoft IIS. For step-by-step instructions on configuring SmarterStats to run with Microsoft IIS 7.0, please refer to the KB article [How To - Set up SmarterStats as an IIS Site](#).

Alternatively, you may run SmarterStats as a virtual directory under an existing site. However, this is not recommended. For more information, please refer to the KB article [How To - Set Up SmarterStats as a Virtual Directory Under an Existing IIS Site](#) .

Note: This help topic assumes familiarity with Microsoft IIS and how it works. SmarterTools recommends using the basic Web server included with SmarterStats if you are unfamiliar with or uncomfortable using Microsoft IIS.

Activating SmarterStats

In order for SmarterStats to function for more than a single site, the product must be activated using a valid license key. In addition, if SmarterStats is moved to another server or assigned to a different database, the product may need to be reactivated.

To access the product activation wizard, click the Settings button in the main toolbar. Then click the Settings navigation pane and expand the Activation folder. Then click Licensing in the left tree view. The edition and license level information for the version of SmarterStats currently being used will load in the content pane.

To activate or reactivate a valid license key, click the Activate or Reactivate button in the actions toolbar. For step-by-step activation instructions, please refer to the KB article [How To - Activate SmarterStats](#) . Note: Activation of a license key requires the server to contact SmarterTools over port 443 (HTTPS). Please ensure that any firewall or internet security software you have installed allows an outgoing TCP port 443 request.

[Return to Getting Started](#)

Upgrading SmarterStats

Because the SmarterStats 6.x download contains all of the installation files needed for any licensing level or edition, upgrading editions or levels is relatively easy. With a valid license key, companies can easily upgrade to the Professional or Enterprise editions or increase the number of agents available in SmarterStats. For more information, see the [Activating SmarterStats](#) . For step-by-step instructions, please refer to the KB article [How To - Upgrade SmarterStats Levels and Editions](#) .

The steps for upgrading to SmarterStats 6.x from an older version of the application vary depending on which version you are currently using. For more information, please refer to the KB article [How To - Upgrade SmarterStats](#) . Note: SmarterStats Enterprise users that have installed the remote service on machines other than the Web interface server should update the remote service before updating the main SmarterStats server.

SmarterStats Deployment Guide

Introduction

Who Should Use This Document

This document is intended for use by all users of SmarterStats to help determine the most effective architecture to gather statistics on their websites and/or in hosted environments.

Determining the Required Architecture

The authors have chosen to divide their recommendations into four categories: individual website deployments, low-volume deployments, medium-volume deployments, and high-volume and specialized deployments. For the purposes of these recommendations:

- Individual website deployments are those in which the Free edition of SmarterStats is utilized to gather statistics on one domain either locally or on a remote, hosted server.
- Low-volume deployments are those where a purchased edition of SmarterStats is reporting on up to 250 websites—based locally and/or on remote servers delivering log files to the SmarterStats server for analysis.
- Medium-volume deployments are those in which a purchased edition of SmarterStats is reporting on up to 2,000 websites from multiple remote Web servers and/or locally hosted websites.
- High-volume and specialized deployments are those in which a higher-level Enterprise edition of SmarterStats is reporting on tens of thousands of websites per Management Reporting Server (MRS) across potentially distributed networks in a variety of methods.

If you have questions about which category may be the best one for your environment, contact sales@smartertools.com for additional information and assistance.

Calculating Disk Space Requirements

SmarterStats uses advanced data storage techniques that enable it to store log files as relational files (called SmarterLogs) that are faster to query and much smaller than the original log files. On average, SmarterStats can store the same log data in 10% to 15% of the hard disk space required by the original log files.

SmarterStats can also export log data to the original log format, eliminating the need to keep raw log files for extended periods of time, resulting in a substantial savings in disk space usage. In most cases, customers actually free up hard drive space on their servers when using SmarterStats instead of taking additional space.

In distributed installations, the MRS is a Web reporting engine that only serves as a request/response engine. It is for this reason that an MRS can handle up to 30,000 websites in a single installation. The MRS itself does not store any imported log data (unless there are sites on the MRS being processed). Log data on distributed Web servers is contained on the remote service installations.

Understanding CPU Requirements

SmarterStats was engineered to process statistics quickly, and to reduce CPU requirements when other processes are running. This allows SmarterStats to run on the Web server without negatively affecting the performance or delivery of the websites on the server.

If the server is dedicated to SmarterStats, or if the server is used strictly as a SmarterStats collector (described herein), the authors recommend allocating additional CPU capacity to the SmarterStats and/or collector processes to increase performance. For instructions on how to make this change, please refer to the Knowledge Base article [How To – Configure SmarterStats to Use More CPU to Process Logs](#) .

Recommended CPU requirements for medium and high-volume deployments are included in this section. However, it should be noted that the following recommendations are estimates based on the average utilization by an average number of sites with an average number of hits, system administrators may need to adjust their CPU requirements based on their site and/or server needs.

CPU Requirements for MRS

MRS fall into three different categories: Web servers reporting on up to 250 sites; high-volume MRS that do not act as collectors and process a maximum of 30,000 sites; and high-volume MRS acting as a collector for a maximum of 1,500 sites and reporting on a maximum of 30,000 total sites.

For Web servers reporting on up to 250 sites, the authors recommend a single-core 1.5 GHz processor.

For high-volume MRS that do not act as collectors and process a maximum of 30,000 sites, the authors recommend a single-core 1.5 GHz processor.

The authors also recommend a dual-core 2.5 GHz processor for high-volume MRS that act as a collector for up to 1,500 sites and report on a maximum of 30,000 total sites.

CPU Requirements for Collectors

The authors recommend a single-core 1.5 GHz processor for collector servers that perform no other tasks besides processing log files from a maximum of 2,000 sites.

Web Servers Using Remote Agents

The authors recommend a dual-core 2.5 GHz processor for individual Web servers running remote agents for up to 1,000 websites.

Individual Websites (SmarterStats Free Edition)

The free edition of SmarterStats can be used to collect statistical data for a single website. In this case, SmarterStats needs to be installed on a user's PC. In the scenario below, the user has a hosted website so the website's log files need to be downloaded via FTP and then imported into SmarterStats. Once the importing process is complete and the SmarterLog files are created, website traffic reports can be generated and viewed via the PC's Web browser.

Remote Web Server and Local SmarterStats Installation

Local Website and SmarterStats Installation

In this scenario, SmarterStats is installed on the same PC or server as the website. This eliminates the need to download the log file via FTP since the website log files are stored locally.

System Recommendations

The authors recommend the following hardware for SmarterStats local machines (PCs):

- 512 MB RAM
- 100 MB hard disk space (only allows for SmarterStats, .NET Framework and small website logs)
- IIS 6* or higher with the Microsoft .NET 4.0 Framework (with all applicable service packs and/or patches)
- Windows Operating System:
 - Windows XP (Home or Professional, SP2 recommended)
 - Windows 2003 or higher
 - Windows Vista
- The Web interface supports the following browsers:
 - Internet Explorer 5.5 and higher
 - Firefox 1.5 and higher
 - Opera 8.5 and higher
 - Mozilla 1.7 and higher
 - Google Chrome

*SmarterStats includes a basic Web server so that the product is fully functional upon installation—even without the existence of IIS or other Web servers. Although not required in single-site and low-volume installations, it is recommended to install IIS 6 or higher in place of the SmarterStats Web server for increased performance and security.

Low-volume Deployment (One Web Server)

Low-volume deployments are those in which a purchased edition of SmarterStats is reporting on up to 250 websites—based locally and/or on remote servers delivering log files to the SmarterStats server for analysis.

SmarterStats Professional or Enterprise Edition

SmarterStats is installed directly on the Web server where it can import the locally stored IIS log files.

System Recommendations

The authors recommend the following hardware for SmarterStats local servers:

- Windows 2003 or higher
- 512 MB RAM
- 1/5 of the total disk space that would be required to store the raw LOG formats
- IIS 6 or higher

Medium-volume Deployments (Multiple Web Servers)

Medium-volume deployments are those in which a purchased edition of SmarterStats is reporting on up to 2,000 websites from multiple remote Web servers and/or locally hosted websites.

SmarterStats Professional with Shares

SmarterStats Professional can import website logs files from Windows IIS Web servers via UNC Shares or FTP server. In addition, SmarterStats can import log files from Linux Apache servers via Samba Shares and FTP server.

System Recommendations

The authors recommend the following hardware for SmarterStats medium-volume distributed networks:

- Windows 2003 or higher
- 1 GB RAM
- 1/5 of the total disk space that would be required to store the raw LOG formats
- IIS 6 or higher

SmarterStats Enterprise Edition

This configuration shows SmarterStats Enterprise edition's flexibility. The imports log files from Linux Apache Web servers via Samba Shares and FTP servers. In addition, SmarterStats receives reports from the SmarterStats Remote Service running on Windows Web servers. MRS

Running the SmarterStats Remote Service on the Windows Web server is the most efficient configuration in terms of speed and system utilization.

System Recommendations

The authors recommend the following hardware for SmarterStats Enterprise medium-volume distributed networks:

MRS

- Windows 2003 or higher
- 1 GB RAM
- 200 MB hard disk space
- IIS 6 or higher

Web Servers Using Remote Agents

- 2 GB RAM
- 1/5 of the total disk space that would be required to store the raw LOG formats
- Microsoft .NET 4.0 Framework (including all applicable service packs and/or patches)
- Windows Operating System:
 - Windows XP
 - Windows 2003 or higher
 - Windows Vista

Note: For ease of installation on Web servers and collectors, an automatable setup package containing just the remote service is available as a separate download.

High-volume and Specialized Deployments

SmarterStats Enterprise Edition with Servers Running in a Collector Role

In this configuration, all collectors are Windows-based servers running the SmarterStats Remote Service. Each collector imports data from other Windows IIS Web servers via UNC Shares or FTP servers. Other collectors can import logs from Linux Apache Web servers via Samba Shares or FTP Server.

This method is highly efficient and all SmarterStats log files are created and stored locally. The MRS is only used as the customer user interface when reports are requested from the collectors and then displayed.

System Requirements

The authors recommend the following hardware for SmarterStats Enterprise high-volume collector networks:

MRS

- Windows 2003 or higher
- 1 GB RAM
- 200 MB hard disk space
- IIS 6 or higher

Collector Servers

- 512 MB RAM
- 1/5 of the total disk space that would be required to store the raw LOG formats
- Microsoft .NET 4.0 Framework (including all applicable service packs and/or patches)
- Windows Operating System:
 - Windows XP
 - Windows 2003 or higher
 - Windows Vista

Note: For ease of installation on Web servers and collectors, an automatable setup package containing just the remote service is available as a separate download.

SmarterStats Enterprise Edition Hybrid

This configuration demonstrates the power and flexibility of SmarterStats in its ability to import and provide website statistics from many different sources.

System Recommendations

The authors recommend the following hardware for SmarterStats Enterprise high-volume networks with multiple data collection methods:

MRS

- Windows 2003 or higher
- 1 GB RAM

- 200 MB hard disk space
- IIS 6 or higher

Collector Servers

- 512 MB RAM
- 1/5 of the total disk space that would be required to store the raw LOG formats
- Microsoft .NET 4.0 Framework (including all applicable service packs and/or patches)
- Windows Operating System:
 - Windows XP
 - Windows 2003 or higher
 - Windows Vista

Web Servers Using Remote Agents

- 2 GB RAM
- 1/5 of the total disk space that would be required to store the raw LOG formats
- Microsoft .NET 4.0 Framework (including all applicable service packs and/or patches)
- Windows Operating System:
 - Windows XP
 - Windows 2003 or higher
 - Windows Vista

Note: For ease of installation on Web servers and collectors, an automatable setup package containing just the remote service is available as a separate download.

Load-balanced Websites (Clustering) for High Traffic Websites

A load-balanced (clustered) website is one in which a single website is distributed across more than one server. Another common term for this type of structure is "Web farm". This technique is utilized by very high-volume websites, websites that have particularly high up-time requirements, and/or other mission critical functions that require fail-over resilience exceeding the norm.

There are two common methods of load-balancing: session-based and hit-based. Each method requires following different steps to ensure SmarterStats analyzes the data correctly. Note: The load-balanced deployment is recommended for users of SmarterStats 4.x and earlier. Because the latest versions of SmarterStats allow importing from multiple log sources, this deployment solution does not apply to users of SmarterStats 5.x and later.

Session-based Load Balancing

Session-based load balancers attempt to keep all traffic for a specific visitor on the same server. This is known as "persistence" or "stickiness."

The following steps explain how to copy and rename logs from per-session load-balanced servers to a single server so that SmarterStats can import them. You can alter or adjust these steps to accommodate particular requirements as needed.

On each of the load-balanced servers, do the following:

- Make a new directory on the C: drive called C:\SmarterStatsLogFileMerge
- Create a new batch file using notepad called DoLogFileMerge.bat that contains the code below (all one line).

```
XCOPY "C:\WINDOWS\system32\Logfiles\W3SVC1\*.LOG" "
\\SmarterStats\Logs\Site1\*.LOG_A " /D /Y
```

Note 1: This code assumes that your original logs on this server are contained at C:\WINDOWS\system32\Logfiles\W3SVC1, that SmarterStats is running on a remote server called file://smarterstats/, and that the share \Logs\Site1 is where the logs should be copied to on that server.

Note 2: The file mask *.LOG_A below should be different for each server. For example, other servers may be *.LOG_B, *.LOG_C, etc.

- Save the file and exit Notepad.
- Go to Start → Programs → Accessories → System Tools → Scheduled Tasks
- Add a new scheduled task.
- Click Browse and click the newly created batch file.
- Choose the frequency of the copy you want. Most will want to set this to Daily, starting around 1:00 AM.
- Enter a username and password for a user that has permission to copy to the share you made.
- Save the scheduled task.
- Test the scheduled task by right-clicking it and choosing Run. After it runs, ensure that the logs were successfully copied.

Hit-based Load Balancing

A hit-based load balancer distributes all requests evenly across its websites. For example, a visitor requesting a Web page may receive the HTML from one server and images from one or more other servers. This is the best performing style of load balancing, but it results in stats that are difficult to track because of time differences and delays in logging the data.

If your website is load balanced and separate log storage locations appear on different servers, you will need to manually combine the logs so that SmarterStats can treat them as one website. To do this, you will need a third-party log combining tool, of which many are available and can be found by searching Google.

Additional information regarding this process and aggregation tools may be available within the Knowledge Base and/or Community Forums accessible through the SmarterTools Portal .

Search Engine Optimization

Users of SmarterStats 5.x and later benefit from the software's search engine optimization (SEO) tools, which help SEO analysts and webmasters monitor SEO campaigns and enhance optimization efforts. SEO processing is performed by the SmarterStats service and each service is limited in the amount of processing it can complete each day.

SEO Processing

In order to retrieve SEO statistics, SmarterStats needs to retrieve search pages from the supported search engines. In order to avoid getting blocked by search engines, page retrievals are limited to one page every 10 seconds. For this reason, the number of page retrievals SmarterStats can do for a specific search engine is limited to six per minute (or 360 per hour or 8,640 per day).

In addition, the number of page retrievals is different for each type of SEO statistic:

- Each keyword added for a site requires 10 page retrievals.
- Each competitor added for a site requires two page retrievals for visibility statistics and one page retrieval for the Google PageRank statistics, resulting in three total page retrievals per competitor.
- Each SEO collection also tracks the user's site, resulting in two page retrievals for visibility statistics and one page retrieval for the Google PageRank statistics (or a total of three page retrievals).

For example, let's assume a SmarterStats site administrator adds two SEO campaigns. The first campaign has four keywords and three competitor sites. The second campaign has six keywords and two competitor sites. The number of page retrievals required is calculated as follows:

Campaign 1				
4 keywords	X	10 page retrievals per keyword	=	40 keyword page retrievals
3 competitor sites	X	3 page retrievals per competitor	=	9 competitor page retrievals

1 local site	X	3 page retrievals	=	3 local site page retrievals

At 10 seconds per page retrieval, it would take 8 minutes and 40 seconds to process the SEO statistics for this campaign.

Campaign 2				
6 keywords	X	10 page retrievals per keyword	=	60 keyword page retrievals
2 competitor sites	X	3 page retrievals per competitor	=	6 keyword page competitor
1 local site	X	3 page retrievals	=	3 local site page retrievals

At 10 seconds per page retrieval, it would take 11 minutes and 30 seconds to process the SEO statistics for this campaign.

In this example, it would take 20 minutes and 10 seconds to process the SEO statistics for the site.

By default, sites are limited to five competitors and five keywords, as a large number of keywords and competitors for a single site could severely limit the processing time available for other sites.

SmarterStats installations with a large number of sites should keep these limits low. However, installations with only a few sites may wish to greatly increase these limits.

As a final note to SmarterStats Enterprise users: Because the processing is done by the SmarterStats service (not the website), an Enterprise version of SmarterStats with 30 remote servers can support 30 times as much SEO processing as a single SmarterStats server.

Summary

The proper configuration and system architecture outlined in this document will provide a solid, reliable foundation. Because variations exist due to different volumes and client needs, the authors suggest starting with these recommendations and then adjusting server proportions, limits, and specifications based on the usage patterns that result.

Automation with Web Services

Who Should Use This Document

This document is intended for use by high-volume and automated businesses environments and hosting companies as they develop procedures to manage their SmarterStats system and work flow. In

addition, this document assumes a basic understanding of Web service technologies and ASP.NET programming.

Automation with Web Services

SmarterStats was built with custom configuration and integration in mind. In addition to being able to customize the look and feel of SmarterStats, developers and/or system administrators have the ability to code to the SmarterStats application using Web services. These Web services allow developers and/or system administrators to automate a variety of different functions and tasks normally available through the product interface.

To view the available Web services and their corresponding functions, go to your Web interface install location and append "Services/SERVICENAME.asmx". These pages can also be used to view the SOAP definition for the Web services.

When you set up a Web reference to these Web services you will want to include the full path to your SmarterStats installation with the service name indicated in each section below (for example, <http://127.0.0.1:9999/Services/SERVICENAME.asmx>).

In the sections that follow, you will find a brief explanation of the available Web services, along with the default installation paths to the specific Web service details page.

Note that some Web service functions may have several numbered forms (ex. AddSite, AddSite2, AddSite3). These are so named to ensure backwards compatibility with legacy systems. For new development, SmarterTools recommends using the highest numbered version.

Note: For a list of constants, refer to the Web Services Constants Reference .

Server Admin Web Service

Services/serveradmin.asmx

These services allow developers and/or system administrators to pull information relating to a server or servers from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). The functions and their corresponding properties are:

- **GetServers** : Returns a list of all servers running SmarterStats Enterprise.
- **GetGlobalDefaults** : Returns the global default settings for a specific server. These default settings include server IP address, server status, default file, document and download extensions, and more.
- **PingServer** : Allows you to ping a specific server and see the results.
- **DeleteServer** : Allows you to delete a server from the Web interface.

- AddServer : Allows you to add a server directly into the Web interface.
- UpdateServer : Allows you to change server configurations on the fly.

Site Admin Web Service

Services/siteadmin.asmx

These services allow developers and/or system administrators to pull information relating to a particular site or sites from a custom page using SOAP, an HTTP GET protocol, or an HTTP POST command (examples for each are provided as well). These functions and their corresponding properties are:

- AddSite3 : Allows you to add sites to the Web interface on the fly.
- AddSiteWithFTP2 : Allows you to add a site with FTP logs directly in to the Web interface.
- DeleteSite : Allows you to delete a site from the Web interface on the fly.
- GetAllSites2 : Returns a listing of all sites being monitored by the Web interface.
- GetSitesForServer2 : Returns sites from a particular Server ID.
- GetSite2 : Returns a single site that you specify and that is listed in the Web interface.
- GetSiteStatus : Allows you to check the site status for a particular Site ID.
- GetRequestedSettings : Allows you to retrieve certain settings for a Site. Requesting a blank list of settings will cause SmarterStats to return settings for all known keys.
- MoveSite2 : Allows you to initiate a move command to the service for a particular site. This would allow you to automate moves within a control panel application.
- SetRequestedSettings : Allows you to set certain settings for a Site. For a list of all keys, request a blank list of settings from GetRequestedSettings.
- UpdateSite3 : Allows you to update information pertaining to a particular site that resides in the Web interface.

Bandwidth Web Service

Services/bandwidth.asmx

These services allow developers and/or system administrators to pull information relating to bandwidth usage for a particular site or server from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). These functions can be extremely useful when providing bandwidth statistics to end users or for bandwidth monitoring and overage billing. The functions and their corresponding properties are:

- GetForSite : Returns all bandwidth for a particular site.
- GetForServer : Returns all bandwidth for a particular server on a per-site basis.

Diagnostics Web Service

Services/Diagnostics.asmx

These services allow developers and/or system administrators to check on the current status of sites from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). The functions and their corresponding properties are:

- `GetSiteStatusForServer` : Allows you to get information about the current import status of all sites on a server.

Statistics Web Service

Services/Statistics.asmx

These services allow developers and/or system administrators to pull information about site activity from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). The functions and their corresponding properties are:

- `GetActivityForServer` : Returns activity stats (hits, visits, views, and bandwidth) for all sites on a server.
- `GetActivityForSite` : Returns activity stats for a particular site.

Global Update Web Service

Services/svcGlobalUpdate.asmx

These services allow developers and/or system administrators to quickly set site settings across an entire installation from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). This may be used to enforce a new policy across a large installation. The functions and their corresponding properties are:

- `ListGlobalUpdateFields` : Returns a list of all fields that can be configured through this web service and any caveats about their parameters.
- `UpdateAllDomainSettings` : Begins the process of setting the specified properties on all sites of the installation.
- `GetGlobalUpdateStatus` : Gets the status of the process that is performing the action previously started in `UpdateAllDomainSettings`.

Product Information Web Service

Services/productinfo.asmx

- `ActivateLicenseKey` : Allows you to activate the license key purchased for your product.
- `GetLicenseInfo` : Returns information pertaining to the current product license.
- `GetProductInfo` : Returns information pertaining to the currently installed product.
- `SetLicenseKey` : Sets a license key for the Web interface.

User Administration Web Service

Services/useradmin.asmx

These services allow developers and/or system administrators to perform various user functions, such as user creation and deletion, updating user information, retrieving user information and more from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). The functions and their corresponding properties are:

- `AddUser` : Automatically adds users to a specific site.
- `SetSiteOwner` : Sets site ownership for a specific site to a specific user.
- `GetUser` : Returns a user for a particular site.
- `ValidateLogin` : Returns login results for a particular user.
- `GetUsers` : Returns all users for a specific site.
- `UpdateUser` : Updates user information for a site.
- `DeleteUser` : Deletes the specified user from a site.

Important Notices

The recommendations in this document reflect the opinions of its author(s) only and are based upon their knowledge and experience. No warrantee or guarantee is expressed or implied as to the efficacy or applicability of the information in this document for a particular environment or application.

SmarterStats, SmarterTools, and their respective logos are trademarks of SmarterTools Inc. All other trade-names and/or trademarks in this document are the property of their respective owners.

Comments and suggestions regarding the contents of this document should be sent to editor@smartertools.com.