

## D2 Sitelock v.2 Quick Start Guide

Follow the steps below to install D2 Sitelock on your web server:

### Platform requirements:

D2 Sitelock uses ASP VBScript and therefore requires a server platform capable of compiling and running ASP 3.0 code. This includes any version of Microsoft Windows with IIS 5.0 (or later) installed (Windows XP, Windows 2000 Professional & Server, Windows 2003 Server Web Edition, Standard Edition and Enterprise Edition).

### Installation:

D2 Sitelock is distributed as a zipped archive in .zip format compatible with compression software such as Winzip, Powerarchiver or Windows XP/2003 built in zip utility. If you require the zip file in another format such as .tar, please contact support at [support@d2computing.co.uk](mailto:support@d2computing.co.uk) quoting the required format.

*Please read the disclaimer and licensing agreement included within the application (license.txt) prior to installing D2 Sitelock.*

1. Unzip the Sitelock\_2\_1.zip file to within your websites root directory, this should create a folder called Sitelock\_2\_1 within your web site. Do not move any of the Sitelock files outside of this directory with the exception of the database file sitelockpro2.mdb.
2. Copy or move the sitelockpro2\_1.mdb database file to a location on your server suitable for storing databases. If using an Internet service providers (ISP's) server to host your site, most of them designate a directory to be used for databases, as this will have the correct permissions already applied to it. If unsure, check your ISP's help files or support for details. If using your own server place the database file in any location you wish, preferably outside of the web root directory for security reasons e.g. E:\databases\sitelockpro2.mdb. You must have the ability to set write permissions on the directory you have chosen, if unsure of how to do this, check with your ISP or if using your own server, view Microsoft's knowledgebase files on setting NTFS file and directory permissions. The database needs the Internet guest account (IUSR\_machinename) to have 'modify' permissions set on it to allow the creation and deletion of data from the database.
3. Open the 'SitelockConn.asp' file which should now be in your website directory under *Sitelock\_2\_1/connections/ SitelockConn.asp* using a text editor such as notepad. This file should contain a line of code similar to the following: **MM\_SitelockConn\_STRING = "DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=E:\webDB\sitelockpro2.mdb;Uid=;Pwd="**. This is the path to your database and needs to be changed according to where you placed the database in step 2. Using the example it may look something like: **MM\_SitelockConn\_STRING = "DRIVER={Microsoft Access Driver (\*.mdb)};DBQ= E:\databases\sitelockpro2.mdb;Uid=;Pwd="** or using an ISP's server it may look like: **MM\_SitelockConn\_STRING = "DRIVER={Microsoft Access Driver (\*.mdb)};DBQ= /db/sitelockpro2.mdb;Uid=;Pwd="**. The path format when using an ISP's server may differ dependent upon the ISP being used, check with your ISP's support team if unsure of the path to use to access your database folder.
4. Assuming the database path is correct and that adequate permissions have been set on the database file then the application is now ready for testing. To test the application, open the Sitelock.asp file using your web browser. The URL will be something like: [http://mysite.com/Sitelock\\_2\\_1/sitelock.asp](http://mysite.com/Sitelock_2_1/sitelock.asp) or [http://localhost/Sitelock\\_2\\_1/sitelock.asp](http://localhost/Sitelock_2_1/sitelock.asp). This should open the Sitelock interface at the login screen; the username/password used to access the application for the first time is d2admin/d2admin. This should be changed

straight away by editing the d2admin account details. Change the username and password of the admin account by clicking the 'edit' link adjacent to the account. It is also recommended you change the admin access level to another level. Important: If you change the admin level, you must also update the access level set in the 'protection/Sladminlock.asp' file within the SiteLock\_2\_1 directory using Notepad or any text editor. This file contains the access level required to enter the admin interface.

**Any errors that occur at this stage are almost certainly to do with either your database path or incorrect permissions set on the database. Check the errors section on the D2 Computing website or the common errors section at the end of this document for details.**

### Configuration:

D2 SiteLock is now ready to configure and use. The first step is to set the variables used by the application. Clicking the 'View variables' link within the top navigation bar opens the variables configuration screen. This should open a screen similar to the one shown below:

Variable	Current setting
Database Path	DRIVER={Microsoft Access Driver (*.mdb)};DBQ=E:\web08\site\lockpro2.mdb;Uid=;Pwd=
<a href="#">Admin Email</a>	paul@durbanfreezer.co.uk
<a href="#">Site Name</a>	SiteLock Test Site
<a href="#">Mail Component</a>	ASPEmail Set
<a href="#">Mail Server</a>	smtp.demon.co.uk
<a href="#">Are users set to expire?</a>	<input checked="" type="checkbox"/>
<a href="#">Days until a users account expires</a>	180 days
<a href="#">Daily report sender activated?</a>	<input checked="" type="checkbox"/>
<a href="#">Default user level</a> (this is the level your 'normal' users will be assigned when registering, this can be either a number or group name. Be careful if changing this once users have already started to register on your site using a previously set level)	60
Number of user records to view per page (to view all records use -1)	20 <input type="button" value="change"/>

Variables are altered by clicking the hyperlink within the left side of the table adjacent to the chosen value. A brief description of the meaning of each variable setting is shown below:

Variable name	Usage
<b>Database Path</b>	This is the path to your database as set in the SitelockConn.asp file, this is not editable from the interface and is shown for information purposes only
<b>Admin Email</b>	This should be changed to an email address that you have access to e.g. <a href="mailto:info@mysite.com">info@mysite.com</a> . This address is used by Sitelock to send information to you such as daily reports
<b>Site name</b>	This is simply the common name for your site e.g. 'UK Motorcycle News website'
<b>Mail Component</b>	D2 Sitelock supports the following email components: CDONTS, CDOSYS, JMail and ASPEmail. Which one you choose depends upon preference and what is installed on your server. Windows 2000, XP and 2003 support CDOSYS by default and Windows 2000 also supports CDONTS by default. The other two are 3 <sup>rd</sup> party components
<b>Mail Server</b>	ASPEmail, Jmail and CDOSYS require that a valid smtp server be specified to relay mail. This is usually your ISP's smtp server such as smtp.demon.co.uk or smtp.yahoo.com etc. Be sure that you have permission to relay through this server. If unsure check with your ISP for their smtp server address. This does not need to be set if using CDONTS as it uses the local server by default
<b>Are users set to expire</b>	If this is set to 'on' then any user registering for an account with your site will have an expiry date set on their account. The date set depends on the number of days you have chosen in the variable below. If this is set to 'off' then the application will ignore any expiry dates set on accounts. Note it is not recommended you use this option if you are using Paypal subscriptions on your site as a users account may expire before their subscription has ended.
<b>Days until a user account expires</b>	This is the number of days a users account is valid for after registration. This will only work if the above variable is set to 'on'
<b>Daily report sender activated</b>	If this is set to 'on' a summary report on user registrations will be sent to your admin email address. The frequency of this email is set by clicking the 'Report sender' link in the top nav bar
<b>Email a user with their username/password after registration</b>	If this is set to 'Yes', users will automatically be sent an email reminding them of the username/password they selected during registration along with a link to your sites homepage
<b>Default user level</b>	This is the default user level assigned to all users who register with your site.
<b>Path to homepage</b>	This is the URL used to reach your site e.g. <a href="http://www.mysite.com">http://www.mysite.com</a>
<b>Custom redirect activated</b>	Custom redirects can be activated if you wish to redirect users of different access level to different pages after login e.g. you can choose to redirect all users of level 4 to page1.htm whereas users of level 5 will be redirected to page5.htm. The actual page locations are set by clicking the 'view custom redirects' link at the top of the variables page.
<b>CC all registration emails to admin account</b>	If this is activated, a copy of all registration emails sent to your users is also BCC'd to your admin address. This is invisible to your users. This feature is useful to track frequency of registrations over a certain period of time.
<b>Force users to validate their account via email before logging in for the first time</b>	If this is activated, when a user registers on your site, an email is automatically sent out to the email address they supplied during registration. This mail message contains a link they must click to activate their account before they can login. The user will only have to click this link once, their account will be marked as active after this. This feature is useful in preventing users registering on your site using a fake email address.

The next set of variables shown below are simply links to the various pages that users are redirected to during registration and login.

Variable	Current setting
Location to send successfully logged in users to:	<input type="text" value="example_pages/example_loggedin.asp"/> <input type="button" value="change"/>
Location to send users who supply wrong username/passwords to	<input type="text" value="example_pages/example_nologin.htm"/> <input type="button" value="change"/>
Page to display if a user attempts to register a username that is already in use	<input type="text" value="example_pages/example_duplicateuser.htm"/> <input type="button" value="change"/>
Page to display upon successful registration	<input type="text" value="example_pages/example_regok.asp"/> <input type="button" value="change"/>
Page to redirect users to if they are not detected as logged in (your login page)	<input type="text" value="example_pages/example_login_page.htm"/> <input type="button" value="change"/>
Page to redirect users to if they try to log on after their account has expired (nb this will only work in you have user expiry switched on)	<input type="text" value="example_pages/example_userexpired.htm"/> <input type="button" value="change"/>

These are changed by overwriting the path to the chosen page and clicking the 'change' button to update the database. The application contains a folder called example\_pages, these are simply examples for testing the application and should be changed to point to pages within your own site. To point to a page outside of the Sritelock\_2\_1 directory use '..' before the path e.g. to point to a login page called login.htm within your own site which is one directory above the Sritelock\_2\_1 directory the path would be ../login.htm, 2 directories above would be ../../login.htm etc.

Once these are set, D2 Sritelock is ready to integrate into your site.

### Creating a login page for your users.

Your site must have a login page where users enter a username/password to gain access. You have to create this yourself, it is simply a standard form on an HTML page which posts its data to the checklogin.asp page in the Sritelock\_2\_1 directory of your site.

Your login form must have two form fields, a username field called SLusername and a password field called SLpassword. This form must post its data to the checklogin.asp file within the Sritelock\_2\_1 directory. A simple example of the HTML code for a form to do this is shown below and an example is included within the application under example\_pages/example\_login\_page.htm (This example also includes some JavaScript to validate your form i.e. make sure users cannot enter a blank username/password)

#### Example login form code:

```
<form name="form1" method="post" action="Sritelock_2_1/checklogin.asp">
<p>username:
  <input name="SLusername" type="text" id="SLusername">
</p>
<p>password:
  <input name="SLpassword" type="password" id="SLpassword">
</p>
</form>
```

There are more examples on the D2 Computing website which include further validation techniques such as adding a 'confirm password' field.

### Creating a registration page for your users.

Your registration page must contain the following form fields (named as below):

- Username field – SLfmusername
- Password field – SLfmpassword
- Email address field – SLfmemail

This form must post its data to the userreg.asp file within the Sitelock\_2\_1 directory. An example of the HTML code for a registration form is shown below and an example is included within the application under example\_pages/example\_registration\_page.htm (This example also includes some JavaScript to validate your form i.e. make sure users cannot enter a blank username/password or invalid email address)

### Example registration form code:

```
<form name="form1" method="post" action="Sitelock_2_1/userreg.asp">
<p>Choose a username:
  <input name="SLfusername" type="text" id="SLfusername">
</p>
<p>Choose a password:
  <input name="SLfpassword" type="password" id="SLfpassword">
</p>
<p>Your email address:
  <input name="SLfemail" type="text" id="SLfemail">
</p>
</form>
```

You should also create pages for users to be shown upon successful registration, attempting to register a duplicate username etc.

### Protecting your pages:

---

**NOTE: Protecting pages requires server side code, this means you can only protect pages with a .asp extension, you cannot protect standard HTML pages.**

There are 2 ways of using D2 Sitelock to protect .asp pages and only allow logged in users of a certain access level to view them.

**1:** You can manually add a line of code to the top of your pages to specify that only users with a certain access level can access them. D2 Sitelock works by assigning a session variable called Session("SLaccess") to a user when they successfully log in. This session variable contains the users access level and remains active until they close their browser or their user session times out after too long a period of inactivity. An example of protecting a page using manually added code is shown below (note – server side code must be placed between <% %> delimiters):

In the example below, the code in blue has been placed at the top of a .asp page to protect it.

```
<%@LANGUAGE="VBSCRIPT"%>
<%
if Session("SLaccess") < 5 OR Session("SLaccess") = "" then
Response.Redirect("login.htm")
end if
%>
<html>
<head>
<title>Example protected page</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
my pages content.....
</body>
</html>
```

This tells the page not to display unless the user accessing it has a valid session variable and their access level is 5 or greater. If they do not meet this criteria, the page will not open and the user will instead be redirected to login.htm. There are various different ways to configure the criteria, the first line in the example above could be replaced with any of these below:

`if Session("SLaccess") <> 5 then` – This only allows users with an access level 5

`if Session("SLaccess") > 5 then` – Only users with an access level of greater than 5

`if Session("SLaccess") <> "" then` – Only authorised users of any level

`if Session("SLusername") <> "johnsmith" then` – Only the user johnsmith can access this page

The last example uses `Session("SLusername")`, this is another session variable set during login and contains the users name they supplied during registration. This can be used in your .asp pages to supply personalised areas of your site e.g. on home.asp:

Welcome back `<% = Session("SLusername") %>` !, this would display 'Welcome back johnsmith !'

If you wish to call a users email address, this is another variable set during login. If you wanted to show the users email address on one of your pages, the code would be:

Hello, `<% = Session("SLusername") %>`, your email address is: `<% = Session("SLeMail") %>`

**2:** The second (and recommended) way of protecting your .asp pages is to use an 'Include' file. An example of this is used to protect Sitelock's interface, notice all the interface pages such a sitelock.asp have the line:

```
<!--#include file="protection/SLadminlock.asp" -->
```

written at the top of the page. This simply takes the code written into the SLadminlock.asp file and writes it into the page upon execution. This does the same job as the code in example 1 but required much less effort and less hand coding. It also has the advantage that if you wish to change the access level of your pages you simply need to change it in one location and all your pages will automatically use the updated access level. An example of the code from an include file is shown below. You can create as many different include files as you wish to protect different areas of your site with different access levels. To create a new one, simply copy one of your existing include files and rename it, changing the required access level.

An example of the code from a protection include file is shown below. In this example, we have a page called softwaredownloads.asp in the root of our site that we want only users of access levels 4487 and 5411278 to be able to access, we want all other users to be redirected to a page called login.htm.

We create a new text file using notepad and name it protectdownloads.asp (remember it needs a .asp extension, make sure you do not have file extensions hidden in Windows and that the file is not actually being named protectdownloads.asp.txt). We then write the following code into this file:

```
<%  
dim D2Isloggedin  
Select Case Session("SLaccess")  
    Case 4487  
        D2Isloggedin = "OK"  
    Case 5411278  
        D2Isloggedin = "OK"  
    Case else  
        Response.Redirect("login.htm")
```

End Select  
%>

We then save the file and this can now be used to protect pages. We put the following line at the top of our softwaredownloads.asp page code and the page is now only accessible to users of access levels 4487 and 5411278:

```
<!--#include file=" protectdownloads.asp " -->
```

### Common error messages

Listed below are some common error messages and their likely causes:

**Microsoft OLE DB Provider for ODBC Drivers (0x80004005)  
[Microsoft][ODBC Microsoft Access Driver]General error Unable to open registry key  
'Temporary (volatile) Jet DSN for process 0x5d0 Thread 0x674 DBC 0x1cb1f5c Jet'.**

This is usually caused by an incorrect database path in the SiteLockConn.asp file. Check this path actually points to the correct location of your database. This error can also occur when incorrect permissions are set on the database, the Internet guest account must be granted read/write and modify permissions on the database.

**HTTP 500.100 - Internal Server Error - ASP error  
Internet Information Services**

Check access permissions on SiteLock\_2\_1 directory and database and if possible restart IIS or application. Sometimes occurs if MS Access lockfile is not removed correctly.

**Server object, ASP 0177 (0x800401F3)  
Invalid ProgID**

Occurs if an email component is chosen that does not exist on the server i.e. if you try to send an email using Jmail when Jmail is not installed on the machine. To resolve, choose another email component.

**Active Server Pages, ASP 0126 (0x80004005)  
The include file 'somefolder/myincludefile.asp' was not found.**

The include file you are trying to use does not exist in the location specified i.e. there is no file called myincludefile.asp located in a folder called 'somefolder'. Check your path actually points to a file.

**Microsoft OLE DB Provider for ODBC Drivers (0x80004005)  
[Microsoft][ODBC Microsoft Access Driver] Field 'SLUsers.SLemail' cannot be a zero-length string.**

A null value has been submitted by either the login or registration form i.e. a user has not supplied one of the required values and submitted a blank form field. Ensure that you have validation in place on your login and reg forms so a user is forced to enter a value. Examples of validation code are found on the D2 Computing website (<http://www.d2computing.co.uk>)

**When testing locally, I cannot log into SiteLock even though a correct  
username/password is supplied, it just keeps going back to the login page.**

This can happen when using personal firewall software such as Zonealarm, try disabling your firewall temporarily and login should succeed. Edit your firewall settings to allow local testing

of asp applications (see your firewall manufacturers support documents for details for how to do this). Also check the account you are trying to login as has the required access level.

### **Users are forced to re-login after a certain amount of time.**

This is by design, if the server detects a substantial period of inactivity from the user (normally 20 minutes) it will end the session and the user will be prompted to login again. It is possible to extent this period if required (search Google for "setting session timeout").

### **Tips to further secure D2 Sitelock:**

Performing the actions below will further secure your D2 Sitelock application:

- **Change the name of the Sitelock\_2\_1 folder.** By default, the zip archive creates a folder called Sitelock\_2\_1 within your web site. It is advisable to change the name of this. Remember to also update your login and registration forms to point to the new folder instead i.e. if you change the name of the folder to 'somefolder' the your login form that previously posted its data to Sitelock\_2\_1/checklogin.asp must now post its data to somefolder/checklogin.asp. No changes need to be made to any of the Sitelock files themselves as they all use relative paths and do not care what the folder they are contained in is called.
- **Locate the database outside of your web root.** If the database file is located somewhere within your web site it is possible for a user to download your database if they know its name and location by simply typing its url into their browser. Always keep your database elsewhere on your server if using an ISP, within their specified database directory.
- **Rename the Database.** It is advisable to rename the database to a hard to guess name such as dr5gc34f.mdb not to an easy one such as users.mdb. Remember to update the *SitelockConn.asp* file with the new name.
- **Use SSL.** If available on your server, use SSL (Secure Sockets Layer) to encrypt the data posted from the login and registration forms. It is beyond the scope of this installation document to provide any support on this. If using IIS, please see the Microsoft Knowledgebase for details of setting up SSL sites.
- **Password protect the Access database.** Using MS Access 2000, it is possible to assign a username and password to the .mdb file so it requires authorisation before allowing a user to open it. For instructions on setting the password see [http://www.microsoft.com/office/ork/2000/three/65t3\\_3.htm](http://www.microsoft.com/office/ork/2000/three/65t3_3.htm) . Once the password is set the *SitelockConn.asp* file needs the username and password you have chosen added to it. If you had set a username of 'john' and a password of 'smith', the connection string within *SitelockConn.asp* would look like: `MM_SitelockConn_STRING = "DRIVER={Microsoft Access Driver (*.mdb)};DBQ= E:\databases\ sitelockpro2.mdb;Uid=john;Pwd=smith"`
- **Use hard to guess access levels for your users.** Rather than assigning users access levels of 1, 5, 10 etc use levels such as 4581249.
- **Change the name of session variables (advanced, but recommended – back up your D2 Sitelock installation before making changes such as this) –** Replace all instances of `Session("SLaccess")` in the D2 Sitelock source code with another session name such as `Session("fkdbcyejstd")`. Remember to also update the lines like `if Session("SLaccess") < 5` etc in your own protection files. Easiest way to do this is to use a program such a Dreamweaver that can do a find/replace throughout the entire application. If you wish to do this using Notepad, references to the session var are found by in the following files: 'addloginrecord.asp', 'adminlogin.asp', 'checklogin.asp' and 'protection/Sladminlock.asp'. Do this at your own risk.

