SiteBeater Integration Manual

Rev. 2.03 – 2004-04-23

SiteBeater

111 Montara Dr.
Aliso Viejo, CA 92656
USA

E-mail: contact@sitebeater.com
Internet: www.SiteBeater.com
Technical Support: support@sitebeater.com

Copyrights & Trademarks

Cover photography by Stacie Ann Smith: http://www.StacieAnnSmith.com/

Disclaimer

JackKnife does not warrant, guarantee, or make representations concerning the contents or applicability of the contents of this manual. JackKnife reserves the right to change the contents of this manual at any time without obligation to notify anyone of such updates.

Send Us Your Comments

If you have comments or suggestions about this manual, please send them to contact@sitebeater.com. A member of the team will acknowledge your mail as soon as possible.

Table of Contents

Introduction

SITEBEATER is the product of several years of development; seeing what was good here, picking up an idea there. After all that time and work, a complete System began to emerge. YOU now own that System

SiteBeater is built upon an API (Applications Programming Interface) that allows for almost unlimited scalability. This manual endeavors to assist you in setting up SiteBeater for day-to-day use, and tries to explain how to implement it for your custom purposes.

The SiteBeater Foundation gives you important tools that let you build whatever entertaining or profitable creation you can imagine: One company used the Foundation to create a self-fulfilling website system where users would purchase websites online using SiteBeater created shopping pages. SiteBeater would then create their web domain and copy the files –their site was up in minutes!

Another company built a single website using SiteBeater. Their site allowed the user to switch between languages and website styles. On their SiteBeater Message Board, users are able to eliminate language barriers as people from across the world post message to one another from a single point.

Still another company decided to use the power of SiteBeater a bit differently. Rather than trying to patch together several software packages, this company used the SiteBeater System to manage all of their accounts, Internet domains, website look and feel, and administer the SiteBeater Systems. Instead of purchasing third party systems and fiddling with them until they kluged into their existing systems, this company simply used the SiteBeater API to create their own Systems. They fit seamlessly into the rest of their site and included administrative functionality, multilingual power, and look and feel manageability.

Imaging what YOU can do with the SiteBeater System. With a small investment, you can reap endless rewards!

The System

The SiteBeater System is separated into several parts. There are two basic parts: the Accounts System and the Domains System – these Systems are known as the SiteBeater Foundation. The rest of the Systems (Image Gallery, Message Board, News and MP3 Catalog.) rest upon those two.

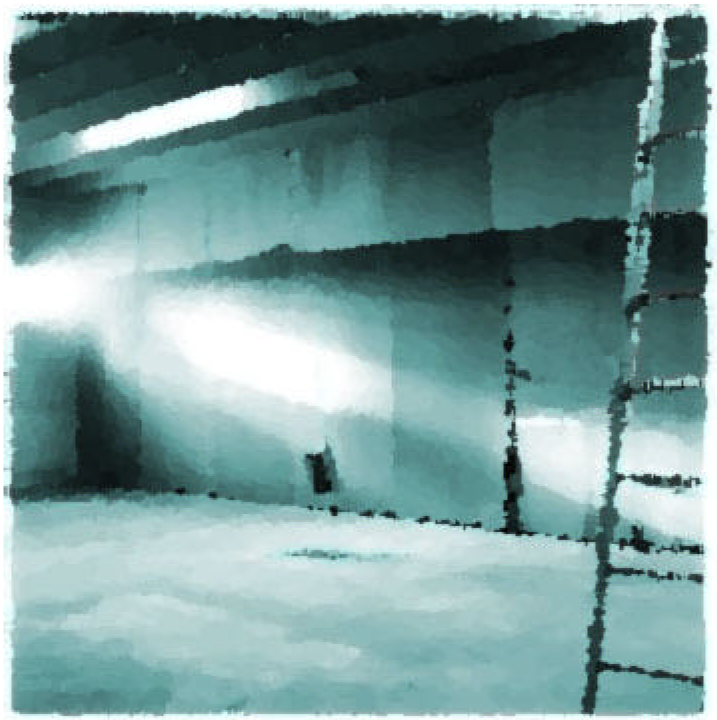
The primary system is the Domains System. There are two parts, or layers, to this system. First is the data. This is all held in the `SB_Domains` database. Second is the ASP code. This is in the `WS_Content\Domains\` directory. What is in the Domains System?

Any structural code or information. The color scheme or text scheme. The Actions (or macros), IIS manipulation, Servers and Configuration Variables are also in the Domains System. Code to access and manipulate this information is also in the list of the Domains System API functions.

Second to the Domains System is the Accounts System. Again, as with the Domains System, there are two parts or layers to this system: the data and the code. The data is within the `SB_Accounts` database and the code is in the `WS_Content\Accounts\` directory. What is in the Accounts System? All account information, including rights and Account Data Variables (ADVs). As well as account Group information. Accounts are only loosely tied to any specific domain. The only ties they have to a domain are the ADVs and the rights they have on specific domains. Any login/password combination can work on any domain if the administrator chooses, but the user may not have the same rights or ADV information from domain to domain. ADVs can be “grouped” with ADVs on other domains so that users can share data across domains.

The MP3 Catalog, Image Gallery, Message Board and other systems also ride upon the Domains and Accounts Systems. Any additional systems you create – including websites – also ride upon the Domains and Accounts Systems. Any System you create can have your custom rights associated with it. It can have color and text schemes specific to it. You can create ADVs that have specific uses in your custom systems. You can also create Groups that have specific rights to your system. Of course you will also want to create Actions for your new system. All these things – and more – are done using the SiteBeater API and web interface.

Installing & Configuring



The SiteBeater Foundation

1

Requirements

Summary:	List of minimum requirements to set up the SiteBeater Foundation.
Audience:	Administrator
Before You Begin:	Review this documentation.
Topics Include:	Third-party Software
See Also:	Edit-On Pro setup. ASPMail, ASPQMail, ASPImage setup.

To set up the SiteBeater System, The following are required:

- MSSQL 2000
- Windows 2000 Server
- Internet Information Services 5.1
- The ability to add a new website DNS and IIS entry.
- A minimum of two MSSQL databases and two MSSQL logins.
- If you are planning on installing SiteBeater on a third party hosting service, or co-location facility, please read the Appendix entitled Installing on Hosting Services

Third-party Software

Other software packages that can be used with SiteBeater include:

- ASPMail or ASPQMail from ServerObjects (www.ServerObjects.com). This is used when a website user requests a login/password reminder email. (CDONTS can be used if ASPMail is not available.)
- ASPImage from ServerObjects (www.ServerObjects.com). This is an optional component used in the Image Gallery System to create thumbnail and Alternate View images.

- Edit-On Pro WYSIWYG editor from RealObjects (www.RealObjects.de). This is an optional piece used in all Systems for editing different pieces of data.
- TqcRunas DLL from The Quimeras Company (http://www.quimeras.com). This is required for administration of domains on multiple servers or across server boundaries.

2

Foundation Setup

Summary:	Follow this checklist to install the Foundation.
Audience:	IT, Developer
Before You Begin:	Review the sections on Requirements, Security Issues, and Database Issues.
Topics Include:	Important Notes
See Also:	Requirements, Security Issues, and Database Issues.

To install the SiteBeater Foundation:

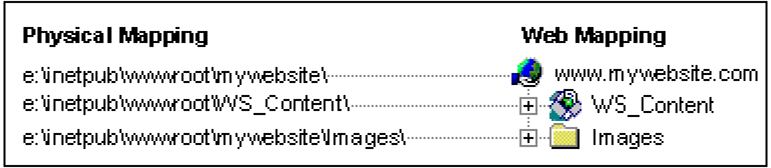
1. Make backups of everything before beginning.

2. Create a website (IIS and DNS entries and directory) if one is not already set-up (for example: www.mywebsite.com in the e:\inetpub\wwwroot\mywebsite\ directory). The online FAQ has an example of how to set up both IIS and DNS.

3. Unzip the SiteBeater code into its own folder on your web server (for example: e:\inteput\wwwroot\WS_Content\).

4. In the website in IIS, create a virtual directory called WS_Content – at the root of your websites directory – and point it to the WS_Content directory from step number 3 that holds all the SiteBeater files. Make sure that the read, write, and script properties are checked. If you cannot create a Virtual Directory, each website using SiteBeater must have it’s own WS_Content directory – with it’s own copy of all of the SiteBeater files.

5. Confirm that the following is now setup: a website (www.mywebsite.com) that points to a directory: e:\inetpub\wwwroot\mywebsite\, that has a virtual directory: \inetpub\wwwroot\mywebsite\WS_Content\, that points to e:\inetpub\wwwroot\WS_Content\:



Physical Mapping	Web Mapping
e:\inetpub\wwwroot\mywebsite\	www.mywebsite.com
e:\inetpub\wwwroot\WS_Content\	WS_Content
e:\inetpub\wwwroot\mywebsite\Images\	Images

(The "Images" directory is an example of another directory in the website and is not required.)

6. Give the /WS_Content/Includes/ directory write access by the Internet user (IUSR) so that the installation script can write to the i_OpenDomain.asp file. The permissions may be restored after the installation is finished.

7. Go to: www.mywebsite.com/WS_Content/SB_Installation/. This will bring up the installation screen. Fill out the database information fields.
    a. **If the MSSQL server is on a <u>different</u> machine than the IIS server**, databases and database logins must be created by hand. There are two databases for the Foundation and they can be any size. The data files may be put anywhere. The databases are called:
        SB_Accounts
        SB_Domains
    Database logins are (by default only – you can change them as you see fit – they are entered into the installation script):
        SB_Accounts – make this the owner of the SB_Accounts database, and make the SB_Accounts the default database. Any password can be assigned (use it in the installation script).
        SB_Domains – make this the owner of the SB_Domains database, and make the SB_Domains the default database. Any password can be assigned (use it in the installation script).
    b. **If the MSSQL and IIS servers are on the <u>same</u> machine**, the administrator may either create the databases by hand (as explained above) or have the installation script create them.

8. Fill out the other fields on the installation script page. Make sure that before you run the installation script that you know:
    a. The MSSQL server name (this is not always the same as the computer name that the MSSQL server is installed on).

   b. The *full* path (including drive letter) to the location where the database files will be created. This is only required if you are having the installation script create the databases for you.
   c. The MSSQL SA login and password. This is only required if you are having the installation script create the databases for you.
   d. The *full* path (including drive letter) to the location of your first website.
   e. The *full* path (including drive letter) to the /WS_Content/Includes/i_OpenDomain.asp file.

9. Finally, return to www.mywebsite.com/WS_Content/ and the SiteBeater system will come up. Login as the administrator (the administrator login and password was entered in the installation script).

10. After the SiteBeater Foundation is installed, remove the installation files from the server (the /WS_Content/SB_Installation/ directory). Other systems may then be installed. The ASPMail, ASPImage, and/or TqcRunas DLLs may also be registered.

Important Notes

Administrators should note that when installing, the script will use the current servers IP address in the first three IP addresses for the first servers ALPHA, BETA, and LIVE (see the section entitled Servers). If possible, it will also use the current server name for the first server. The installation script will not, however, add the new domain to IIS, and therefore administrators will not be able to manage IIS entries for that domain or website using SiteBeater.

The Foundation installation script will create three server entries based on information gathered from the server used to perform the installation (as noted in the previous paragraph). This information does not include several paths, or the administrator login or password for the servers. Before creating any new SiteBeater websites, please enter this path and login information. Go to the SiteBeater administrative pages; Domains->Servers page and click the "more…" link next to each server to add the information.

Review the section on Creating WebPages & Custom Systems to learn how to code the ASP pages in a website. The \WS_Content\SB_Installation_Template\ directory contains a simple example website that you can copy to your new websites directory if you would like to see how to use some of the features in SiteBeater.

Accounts

The SiteBeater Foundation

3

Users

Summary:	Manage users and user settings and rights.
Audience:	Administrator, Developer
Before You Begin:	Decide who will manage users.
Topics Include:	Accounts are website dependent. Login/Logout Manage Users Who can manage users? Editing other users ADV data
See Also:	Rights, ADVs, Initial Rights, and Groups

The *users* are the people who visit each website. This includes customers, managers and administrators.

Manage users by going to:

- any SiteBeater enabled website on which you have rights to manage users
- go to the WS_Content directory
- click on the Accounts link in the top navigation
- click on the Manage Accounts link in the navigation beneath that

Accounts are website dependent.

Each account is created on a specific website. The website on which an account was created is called its "**Home Domain**". Generally, the user can only log onto their Home Domain.

Administrators can change which websites a user can log onto:

- Administrators must have `SB_MA_CHANGE_DOMAINS` rights on that users Home Domain.
- Administrators change which Domains an existing account can log onto by going to the Accounts->Manage Accounts page and clicking the Edit

Accessible Domains link associated with the account they want to modify.
  - From that page, the administrator can allow the user to log onto all Domains, or specify individual Domains that the user can log onto.
  - All users can always log onto their Home Domain.
- Administrators may specify which Domains **new** accounts can log onto by adding those Domains to Domain Groups. See the section entitled SiteBeater Domains to learn about Domain Groups.

Login/Logout

If you have ASPMail or ASPQMail installed and configured (use the `EMAIL_REMOTEHOST` Configuration Variables to specify which email routing server to use), the login page will allow users to retrieve their password by entering their email account. The page will send a letter to the email address associated with their account:

- The body of the letter is contained in the `EMAIL_RETRIEVELOGIN_BODY` text constant.
- The subject is in the `EMAIL_RETRIEVELOGIN_SUBJECT` text constant.
- The from name is in the `EMAIL_LOGIN_FROMNAME` Configuration Variable.
- And, the from email address is in the `EMAIL_LOGIN_FROMEMAIL` Configuration Variable.

The login page also enabled users to turn on “auto-login”. Auto-login sets the login cookies to expire one year from that time. When the user closes the browser, they will not log off of that machine. The cookies set are (this is only for developers reference):

- GAL<DomainUnq> - holds the login
- GAP<DomainUnq> - holds the password
- GAA<DomainUnq – holds the AccountUnq number.
- GAAuto<DomainUnq> - if this is “Y” then they have auto-login on.

The Login.asp and DisableAutoLogin.asp pages (the user is forwarded here when they try to logout and they have auto-login turned on) use these Actions (in the Accounts System):

- `SB_ACCOUNTS_LOGIN_OPEN`
- `SB_ACCOUNTS_LOGIN_CLOSE`

Change these Actions to match the navigation of the website because users will visit these two pages.

Manage Users

Administrators may manage users by going to any website on which SiteBeater has been installed. Go to the /WS_Content/ directory. From there, click the Accounts link, then the Manage Accounts link.

Managing users includes:

- creating new users
- editing account information (see the section on ADVs)
- configuring rights
- changing websites a user can log onto
- moving users to and from Groups.

Accounts are never deleted. For both auditing and customer service reasons, no account can be removed from the system. Accounts are only revoked. Once revoked, accounts can be reinstated with all the original information and rights. Administrators can revoke and reinstate an account as many times as necessary.

Who Can Manage Users?

To view an account on the Manage Accounts page, an administrator must have specific rights on that accounts **Home Domain**. Those rights include:

- revoke (`SB_MA_REVOKE`)
- update login and password (`SB_MA_UPDATE`)
- reinstate (`SB_MA_REINSTATE`)
- create new account (`SB_MA_CREATE_NEW`)
- change domains the account can log onto (`SB_MA_CHANGE_DOMAINS`)

Remember that even being able to *view* accounts is very powerful – the administrator can see logins and passwords.

For an administrator to create a new account, they must have `SB_MA_CREATE_NEW` rights. Of course, if it's available, an administrator can always simply create a new account on the website (if the `ACCOUNTS_SIGNUP` Configuration Variable is set to "OPEN") or by adding one to a Group (see the section on Groups).

If they have `SB_MA_CHANGE_DOMAINS` rights, administrators will see a link next to each account to change the Domains that account can log onto.

A link appears on the same line as each user to the Group(s) they are in. This link only appears if the user is in Groups on their Home Domain.

Tip: To charge for memberships or limit membership

1. keep `SB_MA_CREATE_NEW` and `SB_UG_ADDACCOUNT` rights secure

2. Set the `ACCOUNTS_SIGNUP` Configuration Variable to “CLOSED” This will ensure that the only place accounts can be created on administrative pages.

Editing Other Users ADV Data

Administrators with

- `SB_UD_EDIT` rights can edit other users ADV data
- `SB_UD_VIEW` rights can view other users ADV data

Each ADV entry has radio buttons associated with it labeled “Hidden” and “Visible”:

- Click the “Hidden” radio button to make the element hidden from the general public (it will still be visible to both the user and administrator).
- Click “Visible” to make it visible.

This is useful for things like letting developers know when to add an email address to a Message Board message. If the user has set their email address ADV “Hidden”, then the developer should not display it on the Message Board message.

Next to some of the elements is an asterisk. This means that element is required for the user – but not required for the administrator.

The only ADVs that will appear on this page are ones that the user has entered data into already. Administrators can add data to ADVs that have not been changed by users by clicking the link at the bottom of the page to “Edit this users data – this domain.” It is done this way because on the main ADV editing page, it also lists all of the ADV data from ALL websites that the user has entered data into. It lists **grouped** ADV data (see the section entitled Account Data Variables (ADVs)). The page that appears when you click the “Edit this users data – this domain” only lists the single version of the ADV data for the current domain.

ADVs are sorted and listed by Domain. **Only ADVs on Domains where the administrator has rights to view or edit user data will be listed.** See the section on Account Data Variables (ADVs) for more information about ADVs and sharing or “grouping” ADV data between Domains.

4

Rights

Summary:	Rights give users ability to do things and prevent them from doing other things.
Audience:	Administrator, Managers, Developers
Before Beginning:	Map out who has what responsibilities. Decide who will have power to do what.
Topics Include:	Using SiteBeater rights. Creating custom rights. Using custom rights. Groups & Rights Rights pitfalls. Complete list of Foundation rights.
See Also:	Groups, ADVs, SiteBeater API

Rights tell the System what a user can and cannot do. Typically a user with rights to do many things is called an "Administrator". With SiteBeater, anyone can have rights. There are over 50 different rights already created in SiteBeater that can be assigned to each user. An unlimited number of custom rights can also be created.

View rights by:

- going to any SiteBeater enabled website on which you have rights to manage rights
- then, go to the /WS_Content/ directory
- click the Accounts link at the top
- click the Manage Rights link

A page will appear that has a list of accounts. The administrator must have rights to edit rights (`SB_MR_MODIFY`) on the accounts **Home Domain**. Click on an account to view and modify the users rights.

<u>Administrators can only change rights they themselves have</u>, this affects:

- user rights
- Initial Rights
- Group rights
  - creating Groups (cannot use a Group as a Source Group that has rights that the administrator does not)
  - adding users to Groups (cannot add users to Groups that have rights that the administrator does not have).

Creating Rights

Using rights is simple. Creating rights for custom systems is more involved. The SiteBeater API includes several functions specifically created to deal with rights.

When creating a new system, new rights are usually also created. Each right has an entry in the `SB_Accounts` database, `RightsLookup` table. Each entry consists of:

- `DomainUnq` (the Domain the right is associated with)
- `RightsLvl` (or name of the right)
- `RightsConst` (description)
- `SystemUnq` (the SystemUnq is primarily for sorting purposes – to be able to categorize each right on the management web pages)

To create a new right by hand, add a new entry into the `RightsLookup` table:

1. enter a short description of the right into the `RightsConst` column
2. find the `DomainUnq` of the Domain to add the right to by going to
   a. any SiteBeater enabled website you have rights to view Domains on (`SB_MD_DELETE`, `SB_MD_UPDATE`, or `SB_MD_MODIFY_DOMAIN` rights)
   b. the WS_Content directory
   c. click the Domains link in the top navigation
   d. click the Domains link in the secondary top navigation
   e. the `DomainUnq` will be the number to the left of the Domain name
3. enter the `SystemUnq` this right is to be associated with. If creating the right for a new system, create the system entry first (see the section on Systems). Find the `SystemUnq` by going to:
   a. any SiteBeater enabled website you have rights to view Systems on (`SB_MS_DELETE`, or `SB_MS_UPDATE` rights)
4. enter the name of the right into the `RightsLvl` column. All of the SiteBeater `RightsLvl` strings have the prefix "`SB_`". Avoid appending "`SB_`" to rights names.
5. Or, the SiteBeater API function `ACCNT_CreateRightsLvl()` can be used to create rights on the fly. See the API documentation for the parameter list of the function and an example of its usage.

If the system (that the right is associated with) has already been entered, the new right will appear on the Manage Rights screen. The system is required in order to be able to select it from the drop-down list on that page.

Administrators must also give themselves the new right by adding it to the database before being able to use it. Remember that only users who have a right can manage that right. Administrators can give themselves the right by adding a new entry into the `Rights` table. Each column must be populated with information:

1. This is the `AccountUnq` of the user to get the right.
2. This is the `GroupUnq` of the Group to get the right. Set the `AccountUnq` to 0 if a Group is receiving the right.
3. enter the `SystemUnq` this right is to be associated with. If creating the right for a new system, create the system entry first (see the section on Systems). Find the `SystemUnq` by going to:
    a. any SiteBeater enabled website you have rights to view Systems on (`SB_MS_DELETE`, or `SB_MS_UPDATE` rights)
4. find the `DomainUnq` of the Domain to add the right to by going to
    b. any SiteBeater enabled website you have rights to view Domains on (`SB_MD_DELETE, SB_MD_UPDATE`, or `SB_MD_MODIFY_DOMAIN` rights)
    c. the WS_Content directory
    d. click the Domains link in the top navigation
    e. click the Domains link in the secondary top navigation
    f. the `DomainUnq` will be the number to the left of the Domain name
5. enter the name of the `RightsLvl` from step 4 above.
6. enter the date the right was granted.
7. enter NULL.

Using Custom Rights

Now that the new right has been created, ASP code must be added to the new system to take full advantage of the SiteBeater rights. You can add this ASP code to Actions (see the section on Actions) or you can hardcode it into your ASP web pages. There are three SiteBeater API functions that help developers use rights.

1. The `ACCNT_AnyDomain()` checks to see if the user has the specified right on *any* Domain. Returns TRUE if the user does.
    - **Example:** This function is often used on SiteBeater administrative pages where the administrator can manage several domains. The domains she can choose to manage might be listed in a drop-down list on the page. But, before anything is printed on the page, we must know if the administrator has the right on *any* domain. If she does, we

display the page. Then, we use another API function to display only domains in the drop-down list that she actually has the right on.

2. The `ACCNT_GetDomainW_Rights()` returns the first `DomainUnq` it encounters on which the user has the specified right.
    - **Example:** This is useful if the first website where the user has the right needs to be known. This is often used in combination with `ACCNT_AnyDomain()`. Continuing the example above: the administrator may have the right on several domains. When she first visits the webpage, it must be able to display some default information based on one of the domains the administrator has rights on. It needs to pick just one of those domains, and it doesn't care which domain is chosen.

3. The `ACCNT_SpecificDomain()` function returns `TRUE` if the user has a specific right on a specific Domain. This is the best way to check to see if the user has the right to be where he is, see what he is seeing, or do what he is trying to do.
    - **Example:** Again continuing the example above. Lets say that the administrator is trying to make a change on a Domain other than the current one. Before the system saves the change, it must determine if the administrator has the right to make that change on the other Domain. It can use this function to check the right on the other Domain.

4. The `ACCNT_ReturnRights()` function checks the same thing as `ACCNT_SpecificDomain()`, but only for the current SiteBeater Domain.
    - **Example:** This can be used when determining if songs should be displayed with, or without, a download link in the MP3 Catalog.

The SiteBeater API documentation explains more about all of these functions. The actual SiteBeater code also has some great examples of how to use these functions.

Groups & Rights

Accounts can be placed into "Groups" and these Groups have rights. This means that each user within the Group has the rights of that Group. Users can have both Group and individual rights.

For example, a Group may have rights to view account information on XYZ Domain. If you are in that Group but your account does not have the rights to view account information on XYZ Domain, you will still be able to see account information.

If individual rights are revoked, the user may still have those rights via the Group. To ensure the user does not have a specific right, when revoking rights, double check that the user is not in any Groups with that right. Remove the user from any such Groups or change the Groups rights (remember, when changing the rights of the Group, the rights of all users of that Group will be changed). For more information about Groups, see the section entitled “Groups”.

All API functions that check for rights take into consideration what Group(s) a user is in to see if that or those Groups have the right.

Pitfalls

Some pitfalls and sensitive rights to be aware of:

- Changing Group rights effectively changes all rights for all users in that Group.
- When coding custom Systems, make sure to include checks for rights.
- Be careful about giving out rights to change System settings (on the Domains->Systems page). Rights to change Systems can destroy the SiteBeater installation – this includes all websites using SiteBeater:
    - `SB_MS_CREATE_NEW`
    - `SB_MS_DELETE`
    - `SB_MS_UPDATE`
- Rights to change SiteBeater Domains can also destroy the SiteBeater installation, and more particular, individual websites:
    - `SB_MD_CREATE_NEW`
    - `SB_MD_DELETE`
    - `SB_MD_UPDATE`
    - `SB_MD_MODIFY_DOMAIN`
- Rights to change SiteBeater Domains can give an administrator the ability to manipulate IIS entries. This right can even give them the ability to manipulate IIS entries on multiple servers (if TqcRunas is installed).
- A right to view Servers (on the Domains->Servers page) gives the viewer sensitive administrative NT login and password information:
    - `SB_MSE_CREATE_NEW`
    - `SB_MSE_UPDATE`
    - `SB_MSE_DELETE`
    - `SB_MSE_ADD_DNS`
    - `SB_MSE_ADD_DOMAIN`
- Rights to edit or view other users account information can be an issue if sensitive ADVs such as credit card# or social security# are created.

Associate such ADVs with custom rights (see the section on Accounts Data Variables (ADVs) for information about how to associate an ADV with a right). Rights that will allow an administrator to view user data include:
  - `SB_UD_VIEW`
  - `SB_UD_EDIT`

- If selling a website that uses the SiteBeater System, it is strongly suggested that the rights to change Rights, Groups, Systems, Servers or Domains are not given to the buyer. Charge a nominal fee for making changes related to those requests. Most of the normal day-to-day changes can be done within each System (Message Board, Image Gallery, News, or MP3 Catalog), or with minimal rights.
  - Rights to manage rights:
    - `SB_MR_MODIFY`
    - `SB_MR_VIEW`
  - Rights to manage Groups:
    - `SB_UG_UPDATE`
    - `SB_UG_ADDGROUP`
    - `SB_UG_GROUPRIGHTS`
    - `SB_UG_DELETEGROUP`
    - `SB_UG_ADDACCOUNT`
    - `SB_UG_REMOVEACCOUNTS`
  - Rights to manage Domains, Systems and Servers are listed above.

- If selling a website that uses the SiteBeater System, it is strongly suggested that the right to use ASP in Actions (`SB_MACT_USE_ASP`) is not given to the buyer.

Complete List of Foundation Rights

Rights in this list are given on each website independent of any other website. This means that when rights to do something are given to an administrator on one Domain, that administrator does not have that right on any other Domains. For an administrator to be able to perform the same action on another Domain, she must also be given the right on the other Domain.

Some rights are based on the user they affect. For example, the right to view users logins and passwords depends on which Domain is the users Home Domain. The administrator must have the right to view logins and passwords on the users Home Domain before they can view that users login and password. So, if the administrator has rights to view logins and passwords on the current Domain, but the current Domain is not the users Home Domains, then that administrator will not be able to see that users login and password. A note is given in the description (below) of each right that works this way.

- Management of users:
  - `SB_MA_CREATE_NEW` – this is the right to create new accounts. This allows an administrator to use the SiteBeater administrative pages to create new accounts. This supercedes the `ACCOUNTS_SIGNUP` Configuration Variable. If an administrator has this right, they will be able to create a new account even if the `ACCOUNTS_SIGNUP` Configuration Variable is set as "CLOSED".

  - `SB_MA_REVOKE` – right to revoke accounts. The administrator must have this right on the users Home Domain to be able to revoke the account.

  - `SB_MA_UPDATE` – right to update account logins and passwords. This does not include the right to change any ADV information (*an accounts login and password are not ADVs*). The administrator must have this right on the users Home Domain to be able to update the users login and password.

  - `SB_MA_REINSTATE` – right to reinstate revoked accounts. The administrator must have this right on the users Home Domain to be able to reinstate the user.

  - `SB_MA_CHANGE_DOMAINS` – right to change the SiteBeater Domains an account can log onto. An account can always log into their Home Domain.

- Management of user data (ADV data – not the ADVs themselves):
  - `SB_UD_VIEW` – right to view other users data.
    - This right allows administrators to view users logins and passwords. If data such as a credit card number is added to the list of ADVs, this will also be visible.
    - The administrator must have this right on the Domain on which the ADV data originates. Sometimes ADV data does not originate from the current Domain or the Domain on which the administrator has this right. See the section entitled Accounts Data Variables (ADVs) for more information on "grouping" ADVs to share data between Domains.

  - `SB_UD_EDIT` – right to edit other users data.
    - This includes the right to change a users login or password.
    - Administrators must have this right on the Domain from which the data originates. Sometimes ADV data does not originate from the current Domain or the Domain on which

the administrator has this right. See the section entitled Accounts Data Variables (ADVs) for more information on grouping ADVs.

- Management of rights:
  - `SB_MR_MODIFY` – rights to modify other users rights. The administrator must have this right on the users Home Domain to be able to change their rights. The administrator must also have the right they wish to change before they can change it. (They can either have the right through a Group that they are in, or through their individual rights.)

  - `SB_MR_VIEW` – rights to view other users rights. The administrator must have this right on the users Home Domain to be able to view their rights.

- Management of ADVs:
  - `SB_MADV_ADD` – rights to create new ADVs.

  - `SB_MADV_EDIT` – rights to edit ADVs. If an administrator can edit ADVs, they can also change the rights assigned to any ADVs. **If an ADV is assigned to a right that they do not have, then they can still edit that ADV and change it to a right that they *do* have.** Giving them the ability to access that ADV.

  - `SB_MADV_DELETE` – right to delete ADVs.

- Management of Initial Rights:
  - `SB_IR_UPDATE` – right to change what rights are Initial Rights. Administrators must have a right before they can set it as an Initial Right. The reason for this is so that an administrator cannot set an Initial Right that they do not have, and then create themselves an account to gain the right.

- Management of Groups:
  - `SB_UG_UPDATE` – right to update Group information and users Group specific information. Including – on a user-by-user basis:
    - the number of days to remain on the account in Rotational Groups
    - whether or not an account ignores global rights changes

    Administrators can also change Rotational Group settings (see the Groups section for more information).

  - `SB_UG_ADDGROUP` – right to add Rotational or Standard Groups.
    - When creating Groups, the administrator can add randomly created accounts to them.
    - If an administrator is creating a Standard Group using a Source Group, she must also have *all* of the rights that the Source Group has to be allowed to use it.

  - `SB_UG_GROUPRIGHTS` – right to update Group rights. An administrator must have the right before she can modify it in a Group.

  - `SB_UG_DELETEGROUP` – right to delete Groups.

  - `SB_UG_ADDACCOUNT` – right to add accounts to Groups. An administrator must have *all* rights that the Group has before being allowed to add any account to it. This is so she cannot add an account she has created to a Group that has rights that she does not already have.

  - `SB_UG_REMOVEACCOUNTS` – right to remove accounts from Groups.

- Management of Actions:
  - `SB_MACT_CREATE_NEW` – rights to create new Actions.

  - `SB_MACT_DELETE` – rights to delete Actions.

  - `SB_MACT_UPDATE` – rights to update Actions. Be careful when deciding whom to give delete and update rights. Administrators who do not have `SB_MACT_USE_ASP` rights cannot edit Actions with the `cCODE_DELIMITER` Configuration Variable value in them (if the `cCODE_DELIMITER` is in the Action, it is implied that ASP is in the Action as well).

  - `SB_MACT_USE_ASP` – rights to use ASP in Actions. Without this right, the `cCODE_DELIMITER` value is stripped out of the Action. And, an administrator cannot edit any Action with the `cCODE_DELIMITER` Configuration Variable in it.

- Management of Color Schemes:
  - `SB_MC_NEW_COLOR` – rights to create new colors in any Color Scheme.

  - `SB_MC_REMOVE_COLOR` – rights to remove colors from any Color Scheme.

  - `SB_MC_UPDATE_COLOR` – rights to update the name and value of colors.

  - `SB_MC_NEW_SCHEME` – rights to create new Color Schemes.

  - `SB_MC_DELETE_SCHEME` – rights to delete any Color Scheme. This includes all of the color values within the scheme.

- Manage Configuration Variables
  - `SB_MCONF_CREATE_NEW` – right to create new Configuration Variables. If a user has this right, they can view all of the Configuration Variables including any variables containing passwords or other sensitive data.

  - `SB_MCONF_DELETE` – right to delete any Configuration Variable. Users with this right can also see all of the Configuration Variables, including those containing sensitive data.

  - `SB_MCONF_UPDATE` – right to change any Configuration Variable.

  - `SB_MCONG_EDIT_RIGHTS` – right to change Configuration Variable access rights.

- Management of SiteBeater Domains:
  - `SB_MD_CREATE_NEW` – right to create new SiteBeater Domains. If a website is sold, the seller might be tempted to give the new site owner all rights. If the new owner has this right, they might be tempted to create new SiteBeater Domains.
    - ▪ For an administrator to create a new Domain across server boundaries, TqcRunas must be installed and NT Authentication must be applied to the website from which the administrator is creating the new website (see the sections on SiteBeater Domains and Multiple Servers for more information).

  - `SB_MD_DELETE` – right to delete SiteBeater Domains. Be careful about this one too. Angry employees or customers can tear websites down in minutes.
    - ▪ An administrator may delete Domains on multiple servers if TqcRunas is installed and NT Authentication is applied

to the website from which they are trying to delete the Domain (see the sections on SiteBeater Domains and Multiple Servers for more information).

  - `SB_MD_UPDATE` – right to change the name and domain name of SiteBeater Domains.

  - `SB_MD_MODIFY_DOMAIN` – right to change the grouping of SiteBeater Domains. The grouping of Domains determines which Domains a new account will be able to log onto (see the section on SiteBeater Domains).

- Management of Text Schemes:
  - `SB_MT_NEW_CONST` – right to add new text constants in Text Schemes.
  - `SB_MT_REMOVE_CONST` – right to remove text constants from Text Schemes.
  - `SB_MT_UPDATE_CONST` – right to update the name and value of text constants within Text Schemes.
  - `SB_MT_NEW_SCHEME` – right to create new Text Schemes.
  - `SB_MT_DELETE_SCHEME` – right to delete Text Schemes. This also deletes all text constants within the scheme being deleted.

- Management of Systems:
  - `SB_MS_CREATE_NEW` – right to create new Systems.
  - `SB_MS_DELETE` – right to delete Systems. If a System is deleted erroneously, it could destroy all websites using SiteBeater (see the section on Systems for more information)!
  - `SB_MS_UPDATE` – right to update Systems. If a System is updated erroneously, it could destroy all sites using SiteBeater (see the section on Systems for more information)!

- Management of Mailing Lists:
  - `SB_ML_SEND` – right to send emails to users via mailing lists. Be careful who has this right because whoever has this right can possibly send out spam emails.

  - `SB_ML_CREATE` – right to create new mailing lists.
  - `SB_ML_EDIT` – right to edit existing mailing lists.
  - `SB_ML_ADD` – right to add users to mailing lists.
  - `SB_ML_REMOVE` – right to remove users from mailing lists.

- Management of Logs:
  - `SB_LOG_VIEW_1` – right to view log data from the Accounts System.
  - `SB_LOG_VIEW_8` – right to view log data from the Domains System.
  - `SB_LOG_DELETE_1` – right to delete data from the Accounts System logs.
  - `SB_LOG_DELETE_8` – right to delete data from the Domains System logs.

- Rights associated with Configuration Variables:
  - `SB_MC_CHANGE_EMAIL_METHOD` – rights to <u>change</u> the email method (see the `EMAIL_TYPE` Configuration Variable).
  - `SB_CONF_EDIT_LIVE_DB` – right to <u>edit</u> the live servers DB information. This information is stored in these three Configuration Variables on the LIVE server type (see the section on Configuration Variables for more information about Configuration Variable server types):
    - o `SB_ACCOUNTS_LOGIN`
    - o `SB_ACCOUNTS_PASSWORD`
    - o `SB_ACCOUNTS_SERVER`
  - `SB_CONF_VIEW_LIVE_DB` – right to <u>view</u> the live servers DB constants information. This information is stored in these three Configuration Variables on the LIVE server type (see the section on Configuration Variables for more information about Configuration Variable server types):
    - o `SB_ACCOUNTS_LOGIN`
    - o `SB_ACCOUNTS_PASSWORD`
    - o `SB_ACCOUNTS_SERVER`

- `SB_CONF_EDIT_cCODE_DELIMITER` – right to change the `cCODE_DELIMITER` Configuration Variable.

- `SB_CONF_VIEW_cCODE_DELIMITER` – right to view the `cCODE_DELIMITER` Configuration Variable.

- `SB_CONF_CHANGE_ACCOUNTS_COLORS` – right to change the `ACCOUNTS_COLORS` Accounts System color Configuration Variable.

- `SB_CONF_VIEW_ACCOUNTS_COLORS` – right to view the `ACCOUNTS_COLORS` Accounts System color Configuration Variable.

- `SB_CONF_CHANGE_DOMAINS_COLORS` – right to change the `DOMAINS_COLORS` Domains System color Configuration Variable.

- `SB_CONF_VIEW_DOMAINS_COLORS` – right to view the `DOMAINS_COLORS` Domains System color Configuration Variable.

- `SB_CONF_CHANGE_ACCOUNTS_TEXT` – rights to change the `ACCOUNTS_TEXT` Accounts System text Configuration Variable.

- `SB_CONF_VIEW_ACCOUNTS_TEXT` – rights to view the `ACCOUNTS_TEXT` Accounts System text Configuration Variable.

- `SB_CONF_CHANGE_DOMAINS_TEXT` – rights to change the `DOMAINS_TEXT` Domains System text Configuration Variable.

- `SB_CONF_VIEW_DOMAINS_TEXT` – rights to view the `DOMAINS_TEXT` Domains System text Configuration Variable.

5

Account Data Variables (ADVs)

Summary:	ADVs hold user information. Manage and program with ADVs to maximize your customer service and user interaction.
Audience:	Administrator, Managers, Developers
Before You Begin:	Decide what information to gather from users. Ask marketing what they will need to do their research.
Topics Include:	Creating Custom ADVs ADV Grouping ADV Ideas Programming With ADVs
See Also:	Users section.

Account Data Variables (ADVs) hold information about users. They can hold any text-based information such as names or addresses. An administrator can create as many ADVs as she likes to profile website visitors.

For example, you can create ADVs to hold information about users favorite products to buy on your website, or their favorite vendor or brand. This can help you decide what products to add to your online store. Basic profiling like this help your site visitors feel like you care, and at the same time, you gather important information.

To manage ADVs go to:

- any SiteBeater enabled website on which you have rights to manage ADVs
- go to the WS_Content directory
- click on the Accounts link in the top navigation
- click on the Manage ADVs link in the navigation beneath that

ADVs are visible to administrators with ADV rights to:

- create (`SB_MADV_ADD`)
- delete (`SB_MADV_DELETE`)
- update (`SB_MADV_EDIT`).

Additional rights are needed to edit the data contained in existing users ADVs:

- edit user data (`SB_UD_EDIT`)
- view user data (`SB_UD_VIEW`)

SiteBeater ADVs

The SiteBeater Foundation comes with many built-in ADVs. Those ADVs all have the prefix of "`SB_`".

Special SiteBeater ADVs include:

- `SB_NumLogins` – tallies the number of times a user has logged into a Domain. **Do not Group this ADV with other ADVs from different Domains.** This ADV is hidden from users. Administrators may change the value of this ADV at any time.

- `SB_IPAddress` – stores the last IP address of each user. As with the number of logins ASV, do not group this with ADVs of different Domains.

Creating Custom ADVs

To create a new ADV:

- go to any SiteBeater website
- go to the Accounts section
- then the Manage ADVs page
- click on the "New" link at the top to go to the creation page.
- Fill out the information required for each ADV:
    - Give your new ADV a name. Names are unique to each SiteBeater Domain. The ADV name is used when programming ADVs. The `ACCNT_WriteADV()` and `ACCNT_ReturnADV()` API functions both require the ADV name as a parameter.
    - Enter a description. This description is what is usually displayed on the webpage next to the ADV form field. For example, an ADV may have the name of `ADV_CCName` and the description of: "Name on credit card".
    - Choose to make it <u>required</u> and/or <u>visible</u>.
        - If ADV is required, then by default, it is also visible.
        - Visibility of an ADV is different from "Visible" and "Hidden" on the user data form. Visibility of an ADV means that it is dynamically displayed on the user data

        form (the “Preferences” page) when all other criteria are met (rights and system).
        - Hidden ADVs are used when administrators might want some data hidden from the users – such as the users IP address or last date they visited the site.
    - Choose to make it a text field or notes field.
        - The text field holds less than 250 characters.
        - The notes field holds nearly an unlimited amount of text data.
        - Switching between the two will blank the original field. For example, if an ADV was a text field and it is changed to a notes field, the next time the user edits that data (or sometimes even a form with that data on it), the text data will be erased. If the ADV is switched back to a text field before the user ever edits it again, the original data will remain. In other words, the data is changed upon editing (if an administrator changes an ADV by mistake, they can quickly change it back and possibly avoid loosing any data).
    - Optionally associate the ADV with a System.
        - Associating ADVs with a System provides a method of categorizing ADVs.
        - Some ADVs don’t make sense in some Systems. For example, the SiteBeater Message Board comes with several ADVs that do not show up on the users Preferences page. The Message Board ADVs only make sense in the message board. Things like board sort order or message signature only apply within the context of the message board.

ADV Grouping

ADV data can be shared across SiteBeater Domains. ADVs that share data across Domains must be added to an ADV Group. ADV Groups consist of a:

- single ADV from one Domain grouped with a single ADV from one or more other Domains.
- Each group can only have one ADV from a domain in it at a time.

Data is bridged across domains through grouped ADVs. For example, an administrator may group the `SB_FirstName` ADV from domain1 and domain2. Users who can log onto both domains can enter their first name on domain1 and see it on domain2 without having to ever enter it on domain2.

ADV Groups can cause problems. Here are three example situations that can occur and possible solutions:

1. If an administrator adds an ADV to a group, users may suddenly see data on websites they did not enter that data on. For example, the administrator has given a user rights to log onto domain1 and domain2. That user enters their first name on domain1 but not on domain2. Then the administrator groups `SB_FirstName` from domain1 and domain2. If the user goes to domain2, they will suddenly see their first name on that domain – even though they never entered there!
   - **Solution:** Unless properly informed, users may worry that the sites are not secure, or that they are sharing data that they didn't authorize to be shared. Make sure it is plainly clear to users that data is shared across several Domains. A good marketing method is to tout this feature as one of the benefits of your website.

2. This time, the administrator removes an ADV from the group. For example, the administrator has already grouped `SB_FirstName` on domain1 and domain2. The user enters their name into domain1 and sees it on domain2. If the user never changes their name on domain2, the system will always use the data from domain1. Later on, the administrator decides to take `SB_FirstName` from domain2 out of the ADV group. Now, when the user goes to domain2, their first name has suddenly disappeared!
   - **Solution:** As in the first example, users may worry about site integrity. It may appear that domain2 has lost data. Make sure the users understand that the websites are sharing data. Also, try and plan ahead so you never have to remove an ADV from a group. Make a general announcement using Mailing Lists to let all of your users know that a change is about to occur and the nature of the change.

3. Choosing which ADV data to use. Finally, if a user adds their first name to both domain1 and domain2, they will have two sets of data. They could have entered "Fred" on domain1 and "Fred Jr." on domain2. What happens if the administrator then adds the `SB_FirstName` ADV from domain3 to the group? The system will check the data from domain1 and domain2 and populate domain3 with the version of data that was changed last. Lets say the newest data is "Fred Jr." from domain2. Domain3 will then reference (it is not copied over) "Fred Jr." from domain2 until the user changes their first name on Domain3 or, the first name ADV from Domain3 is removed from the group.

Consider adding strategic marketing text on the website to let users know that their information is shared across several websites, the benefits of shared data, and that they can use their account on several websites.

To change ADV Grouping:

- go to the Manage ADVs screen
- click the Modify Grouping button.

- This page lists all existing ADV Groups.
- Beneath the group listing, the administrator may choose an ADV to add to a Group.

ADV Ideas

Any user information can be stored in ADVs. There are an infinite number of systems that use ADVs can be created, and ADVs that store answers to questions that can be asked users. Ideas include:

- Create several online stores on different SiteBeater Domains. Use ADV information as an “Internet Wallet” and website-spanning shopping cart. The users would add items from one store to their cart, then go to another store and add items to their cart from that store. Because all the stores are using the SiteBeater System, the users information is website independent and can be used on any store. Then, they can checkout from any of the stores.
- Allow users to choose the Text Scheme (language) and Color Scheme of the website and save it in their preferences.
- Add website specific information. With a site about sports cars, let users add information about their car. Or with a website about gardening, let users enter information about the plants they grow. Or let them keep a list of favorites.
- Keep statistical information of users with hidden ADVs that the websites code populates and updates. Save information like last date the user logged in, or number of visits, or pages visited.
- Ask users what they like and display banner-ads based on their likes.
- Use Crystal Reports® or other software to create reports of the data stored in ADVs.

Programming With ADVs

There are three API functions that deal with ADVs:

- `ACCNT_WriteADV()` writes data to an ADV
- `ACCNT_ReturnADV()` returns ADV data
- `ACCNT_ReturnVisibility()` lets developers know if the user wants to hide from general display the ADV data.

Users can set ADV information as “Visible” or “Hidden”. For example: users may want to display their email address with their messages on the Message Board. Use `ACCNT_ReturnVisibility()` to determine if the user wants it shown (it returns “PUBLIC” if they want it shown). Use `ACCNT_ReturnADV()` to get the email address data from the ADV and add it to HTML to create a MailTo link. If the user does not want it displayed, change your HTML so that it does not have a MailTo link.

Using `ACCNT_ReturnADV():`

- When developing custom systems, or using ADVs in Actions, the `ACCNT_ReturnADV()` API function will only return the ADV data if the current user is the owner of the ADV, or if the ADV has been set as Visible by the owner of the ADV.

- Sometimes you need to read the ADV data even if it is hidden to the general public. One example of this is when a user sends an email to the administrator in the Message Board using the “alert administrator of a offensive post” feature. What happens if the administrator sets their email address as a hidden ADV? If we cannot get the administrators email address, the alert email goes nowhere or returns an error! Even though the administrator is not the user logged in, we still need their email address to send the mail. Use the “OVERRIDE” keyword in the `ACCNT_ReturnADV()` function in place of the `sViewLvl` parameter to have it return the value of the ADV even though that ADV is supposed to be hidden.

- Don’t forget to include the `SystemUnq` (the System’s ID number) of the ADV you wish to get the data from. If the ADV is associated with the Image Gallery System, you cannot ask for it using the Message Board `SystemUnq.` To find the SystemUnq:
    - make sure you have rights to view System data (`SB_MS_DELETE, SB_MS_UPDATE`)
    - go to any SiteBeater enabled website
    - go to the WS_Content directory
    - click the Domains link in the top navigation
    - click the Systems link in the next level of navigation
    - the `SystemUnq` is listed to the left of the System in the “System ID” column.

- If the ADVs type is short text (or `VarChar`), you must specify in the parameters that you need the short text returned. If the ADVs type is long text, then you need to specify in the parameters that you want long text returned. If you ask for the wrong type, you will either get nothing back, or the wrong data.

- Make sure that those who are using the API functions communicate with those who are creating and editing the ADVs so that no changes to either the ADV System or ADV Type happen without the developers also changing their function calls.

See the API for more information about the functions used to read and write ADVs. The API also lists all of the possible parameter values.

Note: if an ADV is not in an ADV Group, the `GroupUnq` for it in the AccountMap table, SB_Accounts database, will be a 0. Developers should check for this in code that requires information about *ADV grouping* or when trying to return ADV data without using the API functions. This should only be a concern if developers plan on querying the database by hand. Developers must remember to get the correct ADV data from the Group that the ADV is in and the *Domain* the user is currently on. This sort of query is <u>very advanced</u> and we suggest you use the API functions instead. If the ADV is not in a group (the `GroupUnq` is 0), then developers do not need to worry about ADV Grouping.

6

Initial Rights

Summary:	Initial Rights give new members the power to do specific things.
Audience:	Administrator, Managers
Before You Begin:	Decide what rights new members should have.
Topics Include:	Scenarios
See Also:	Users, Rights

It is often useful to give new users a limited set of rights to begin with. Set rights on the Initial Rights pages to what new users should have for the Domain. All new accounts will then be given these rights.

Administrators must have the right to be able to set it as an Initial Right.

To manage Initial Rights go to:

- any SiteBeater enabled website on which you have rights to manage Initial Rights:
  - `SB_IR_UPDATE` – update Initial Rights (you must have the right to be able to change it)
- go to the WS_Content directory
- click the Accounts link in the top navigation
- click the Initial Rights link in the navigation beneath that

Scenarios

- If a website has a pay-to-join setup where users can download MP3s, set the Initial Rights to allow all members the right to download (or listen to) MP3s.
- Do the same sort of thing for an online Image Gallery – give all new members the right to create new galleries.

- Give administrative rights to all new members on a site closed to the public.

7

Groups

Summary:	Groups are a powerful tool to manage account rights.
Audience:	Administrator, Managers, Developers
Before You Begin:	Decide what Groups are needed, who will go in each Group, and what rights to give the users in each Group.
Topics Include:	Standard Groups Rotational Groups Programming With Groups Uses for Groups Pitfalls Random Account Creation
See Also:	Rights, Users

SiteBeater Groups combine user accounts and the rights assigned to those accounts. Many times different accounts have the same rights. Rather than change rights on an account-by-account basis, a Groups rights may be changed, all accounts within the Group will be affected. There are two types of Groups: Standard and Rotational.

Manage Groups by going to:

- any SiteBeater enabled website on which you have rights to manage Groups:
    - `SB_UG_UPDATE` – change existing Group settings
    - `SB_UG_ADDGROUP` – create new Groups (you must have all the rights of a Source Group if you are using one)
    - `SB_UG_GROUPRIGHTS` – change Group rights (you must also have the right to be able to change it)
    - `SB_UG_DELETEGROUP` – delete existing Groups
    - `SB_UG_ADDACCOUNT` – add accounts to Groups (you must have all rights within the Group you are adding the account to)

  - `SB_UG_REMOVEACCOUNTS` – rights to remove accounts from Groups
- go to the WS_Content directory
- click on the Accounts link in the top navigation
- click on the Groups link in the top navigation beneath that

Standard Groups

Within Standard Groups, changing the rights of the Group changes the rights of all accounts within it.

- Users can be in multiple Standard Groups at the same time.
- A user in several Standard Groups will gain all rights for all the Groups they are in.
- Users can also be in Standard Groups on different SiteBeater Domains. When this happens, they gain rights on all the SiteBeater Domains in which they are Group members.

Rights of each individual within Standard Groups can still be changed. Only *giving* such a user additional rights will have an effect. Revoking individual member rights will not prevent the user from having the rights because of the Groups they are in.

There are two ways to completely revoke individuals' rights when the user receives the rights from a Group:

- the administrator must remove the user from that Group giving them the rights
- or revoke the rights from the Group

The SiteBeater Foundation comes with the "SiteBeater Owners" Standard Group. This Group has *all* rights. When any additional SiteBeater Systems (Message Board, Image Gallery, etc) are installed, *this Group is given all rights for the new System*. The first account created when installing the Foundation is automatically added to this Group.

When a Standard Group is deleted, it leaves all the accounts within it intact in the system, but those accounts loose all rights associated with the Group deleted.

Rotational Groups

Sometimes, it is advantageous to have the ability to change accessibility on a scheduled basis. A Rotational Group allows administrators to create accounts that are active for a set number of days.

These Groups contain accounts that are only active for a few days at a time.

- Administrator may set the specific number of days for each account, or for the Group as a whole.
- Administrators may have the system randomly generate accounts when the Group is first created, or create new accounts later on. No previously existing accounts can be added to this type of Group (because of the treatment of the account within the Group – see the paragraph below explaining what happens to accounts within Rotational Groups).
- Group rights may be set.
- Individual accounts rights may be set.
- Accounts may only be in one Group at a time if they are in a Rotational Group.
- Administrators may specify how many times to "rotate" through the Group. Or, the Group can rotate forever. Rotation means restarting again with the first account once the last account has expired.

Rotational Groups were created with the assumption that they would protect certain parts or functions of a website. Users would be given access to these parts for a limited amount of time. For example, company A sells web services. They want to give potential customers the ability to test their service for a day. The company would then create a Rotational Group that changes login and password each day. This Group has 365 accounts in it. Then when a customer requests a demo, the company representative gives them the login for that day. The customer then only has rights to view the website for one day. The Group does not "rotate" for a whole year.

Rotational Groups contain a slightly different type of account. Technically speaking, the accounts within Rotational Groups are the standard accounts. SiteBeater simply handles them differently. For example, the System does not allow users logged in with an account in a Rotational Group to edit account preferences. (You can customize the System and allow users to change preferences. But, it may be odd to have an account – complete with email, address, etc. – become disabled after three days, only to be re-enabled again after six months.)

Deleting a Rotational Group deletes all accounts within it. It does not simply remove them from the Group. This is the **only** time any account within the system can be deleted.

Groups & Rights

Rights affect Groups in ways that may not be initially clear. Administrators cannot give rights that they themselves do not have. It is fairly obvious, then, that administrators cannot give a Group rights that they do not have. Not so obvious is the fact that administrators may not create new Groups using Source Groups that have rights that they (the administrator) do not have. The reason is that an administrator could create a new Group and randomly generate an account in the new Group. The administrator would know the new accounts login and password, and then be able to log in as that user with their new Group rights.

Also, administrators cannot add users to Groups that have rights that they do not have. As with the previous example, an administrator could create a new account in a Group that has rights that they do not have. Once the account has been added to the Group, the administrator could log in as that user to gain those additional rights.

Programming With Groups

With the SiteBeater API, developers can create custom rights (see the section on Rights). Rotational Groups and Standard Groups could be given those custom rights. Then developers could set areas or functions on the website as accessible only to users in those Groups.

Remember, the functions dealing with rights in the SiteBeater API are all Group compatible.

Uses for Groups

- **Example 1:** There are three password protected areas on the website. One area for potential merchants, another for potential retailers, and a third for potential investors. All three types of people come and go over time. Some way is needed to give all these people access to their respective areas for a limited time. Think of the difficulties of having to worry about – say: 4 retailers on Monday, 8 merchants Tuesday, and 3 investors Wednesday. What a pain to track when each of their accounts expires. Plus, all those accounts will expire and clutter the system. Use Rotational Groups to solve this problem. Create three Rotational Groups – one for each type of person – retailer, merchant, and investor. Then, give each Group their own custom right to access their particular areas on the website. When anyone from any of the three groups of people call, give them today's login and password for their group.

- **Example 2:** Developers use SiteBeater to program a high security website. Create a custom right and use the SiteBeater API to apply it to all the web pages. Then, create a Rotational Group and add the custom right to it. The login and password will change constantly and prevent the password from leaking.

- **Example 3:** Using the same scenario above, the company might have a high turnover rate. With the account to access the site constantly changing, past employees are prevented from knowing the login and password. And the administrator won't be required to remember to change all the passwords.

- **Example 4:** Standard Groups are great for their ability to help manage many different types of administrators or managers. For example, one Group can only change Actions, another can only change text in Text Schemes (call that the Translator Group), another can only add new SiteBeater Domains (maybe a Purchasing Group), etc.

- **Example 5:** A company has a pay-for-download website that sells four levels of membership subscriptions. The Image Gallery has four different rights that users must have to download each of the four Alternate View images associated with images in the gallery. The company could make four Groups. The first Group would have the right to view the first Alternate View image. The second Group would have the rights to view the first and second Alternate View image, and so on.

  Since administrators may also associate Configuration Variables with Groups, the company could also set a download limit for each Group. The administrator would simple need to create a copy of the `IMAGEGALLERY_INITIAL_NUMDOWNLOADS` Configuration Variable for each of the four Groups. Then members of the first Group could download 10 images each day, the second Group 20, and so on.

Pitfalls

- Administrators who have rights to add users to Groups are, by default, given rights to create new accounts. The *only* way to add new users to Rotational Groups is by creating them. The form for adding new users creates these accounts automatically before adding them to the Group.

- Accounts in Rotational Groups will time-out. If users question why the account they were using does not work anymore, be ready to explain that it timed out after X days.

- Accounts in Rotational Groups do not generally have ADVs. There is usually no user explicitly associated with the accounts in Rotational Groups. However, linking those who login with accounts in Rotational Groups to the user data edit page (the Preferences page) is possible. It is just not recommended. Remember too that there is no way to move ADV data from one account to another.

  ADV information will be stored with accounts within Rotational Groups from the various Systems. For example, the hard disk space limit of the Image Gallery will be stored as an ADV for accounts within Rotational Groups just like any other account.

- It is possible to add users to Groups when the users do not have access to log into the Domain of the Group (see the section on Users for more information about controlling which Domains a user may log onto). The reason is that when a new Domain is created, it copies all Groups and users who are in those Groups – even if those users do not have access to log into that new domain – to the new Domain. Administrators can give them access to log onto the new Domain, or remove them from the new Domains Group later on. Be careful who is added to the Groups – make sure they have access to log into the Domain.

Random Account Creation

Accounts can be generated when creating a new Standard or Rotational Group. These accounts will have all the rights from the Initial Rights (see the section on Initial Rights) for the Domain, and will be able to log onto all the SiteBeater Domains in the Domain Group (see the section on SiteBeater Domains).

The logins and passwords of these accounts are generated randomly from a source file in the `\WS_Content\Accounts\Groups` directory called: `Words.txt`. This file has 100,000 English words in it, one on each line. The algorithm used to gather logins and passwords expects 100,000 entries to be in that file. If there is less, it has a chance of returning an error, if there are more, the ones past 100,000 will be ignored. **Some of the words may not be appropriate for every situation**. If any of the words are changed or removed, please remember to keep the number of entries at 100,000.

8

Mailing Lists

Summary:	Send emails in bulk to SiteBeater accounts with mailing lists.
Audience:	Administrator
Before You Begin:	Make sure SiteBeater is able to send emails.
Topics Include:	
See Also:	Configuring ASPMail, ASPQMail & CDONTS

SiteBeater Mailing Lists allow an administrator to send out bulk emails to users in the SiteBeater Accounts database. An administrator may create as many lists as she likes. Each list can contain an unlimited number of accounts, and each account can be in an unlimited number of mailing lists.

Manage Mailing Lists by going to:

- any SiteBeater enabled website on which you have rights to manage Mailing Lists:
  a. `SB_ML_CREATE` to create new lists
  b. `SB_ML_EDIT` to edit existing lists
  c. `SB_ML_DELETE` to delete lists and remove all accounts from the list
  d. `SB_ML_ADD` to add users to lists.
  e. `SB_ML_REMOVE` to remove users from lists.
  f. `SB_ML_SEND` to send emails to the users on any list.
- go to the WS_Content directory
- click on the Accounts link in the top navigation
- click on the Mailings link in the navigation beneath that

Sending Emails

When sending emails through a mailing list, make sure that your email routing server allows for multiple emails to go through it from the domain from where you are sending the emails. Often times the email server will either completely disallow emails from the domain, or only allow one at a time (one from each page hit). Remember that the email routing server is set in the `EMAIL_REMOTEHOST` Configuration Variable. There are three versions of this variable on ALPHA, BETA, and LIVE servers.

Emails may be sent as HTML or plain text. The administrator has the option of sending a test email to herself. This is a feature that should be taken advantage of to make sure the email looks like it is intended to. The test emails are sent to the administrators email address – make sure one is entered (in the Preferences page).

The emails will be sent to each user using the accounts first and last names as entered in their Preferences page. It will also use the email address from the Preferences page. The Mailing List uses the "`OVERRIDE`" statement in the `ACCNT_ReturnADV()` function to force it to return this information in the event that the user set this data as "`PRIVATE`".

To prevent the web page that sends the emails from "timing out", the page will only send 50 emails at a time. It will then pause for five seconds before sending another batch of 50. It will continue like this until all of the emails have been sent. Do not close the browse window until the page displays the Success message.

After an email is sent via a mailing list, the email data is logged. The Subject, From Name, From Email, and Body, are all stored in separate log entries. Only one copy of the email is saved in the logs.

Users may "opt-out" of each list on the website from their Preferences page. Only lists that they have been added to appear on their Preferences page.

9

Polls

Summary:	Create and manage polls so your users can vote on questions you ask.
Audience:	Administrator
Before You Begin:	
Topics Include:	Managing Polls Managing Answers The Poll Action Other Poll Pages
See Also:	Actions, Creating WebPages & Custom Systems

SiteBeater Polls allow administrators to let their site visitors submit feedback. Not only do the polls allow users to answer the questions, but users can also post their thoughts as comments regarding the poll if the Message Board System is installed.

Note: The polls use the Message Board System files. Make sure the /WS_Content/MsgBoard/Includes/ directory exists before trying to display a poll (this directory and the files within it should have been included with the Foundation files or the Message Board files if you purchased that System).

Manage Polls by going to:

- any SiteBeater enabled website on which you have rights to manage Polls:
    a. `SB_P_CREATE` to create new polls.
    b. `SB_P_EDIT` to edit existing polls.
    c. `SB_P_DELETE` to delete polls.
    d. `SB_P_ADDANSWERS` to add new possible answers to polls.
    e. `SB_P_DELANSWERS` to remove possible answers from polls.
    f. `SB_P_EDITANSWERS` to modify the possible answers in a poll.
    g. `SB_P_RESET` to reset the tally of the number answered for each question

- go to the WS_Content directory
- click on the Accounts link in the top navigation
- click on the Polls link in the navigation beneath that

Managing Polls

For an administrator to create a new poll, he need to first create the question. Create questions by:

- Choose an optional Message Board Conference and Thread (this option only appears if the Message Board System is installed). You can find both numbers in the Querystring when you go to the thread in your browser. The `iConfUnq` is the Conference number, and the `iThreadUnq` is the thread number. **You must create the conference and thread beforehand**.

- If desired, check the checkbox to allow nonmembers (users not logged) to answer (or even see) the poll.

- Select a start and end date. The poll will not show up if the current server date/time is outside of the date/times you choose here. Also, if when you add the poll code to your page (see the section on The Poll Action below), you don't declare and the `iQuestionUnq` global variable, the system will choose polls to display from those that are active based on start and end date.

Managing Answers

Once a poll has been added, you can return to the main poll page and add answers by clicking the "Edit" link next to the poll you wish to add answers to.

- Clicking the "Edit" link next to the poll will take you to the poll editing screen. At the top (if you have rights) will be a link to "Edit Answers". Click this.

- From the Poll Answers page, you can add as many answers to the poll question as you like. Remember that these answers must fit in the narrow poll box.

- Add an answer.

- Once you've added answers, you can reorder them by clicking the up and down arrows next to them in the "Order" column.

- You may also delete them.

- If you have `SB_P_RESET` rights, you will also see a "Reset" link. Clicking this will reset the number of times this answer has been chosen to 0.

- You can come to this screen at any time to view the number of times each answer has been chosen, or edit the answer text.

The Poll Action

The Poll Action is what displays the small poll table with the questions in it.

Developers can customize the SB_POLL Action to change the actual poll box that is displayed for the end users. A good way to edit this Action is to copy all the code into a test.asp file, and replace the %PW% with opening and closing ASP tags (the <% and %> tags). Then you can go to (your website) www.mysite.com/test.asp to see how your changes are coming along. When you are finished, replace the <% and %> tags with the %PW% tag again.

REMEMBER that in Actions, you cannot break up code blocks into separate %PW% blocks. (For example, if you have an if/then statement, you cannot have the "If (1 = 2) Then" in one %PW% block, and the "End If" in another %PW% block, both must be in the same %PW% block. This is why the whole SB_POLL Action is within a single big %PW% block and all HTML is output via Response.Write statements.)

Adding Polls to The Website

The Poll requires that you code the SB_POLL Action into whichever page you would like it to appear. The /WS_Content/Public/Polls/Default.asp page illustrates a simple example of how to do this. You can call the Call DISP_Action("SB_POLL") subroutine at any position in your HTML by opening an ASP block and calling it. This allows you to position the poll box anywhere on the page.

Follow these steps to code the SB_POLL Action into your page:

- Include the /WS_Content/Includes/i_Includes.asp file in your page. You can use this code at the top of the page before any ASP: `<!--#INCLUDE VIRTUAL="/WS_Content/Includes/i_Includes.asp"-->`

- If you want to specify which Poll to display, you must declare globally the `iQuestionUnq` variable (within ASP delimiters): `Dim iQuestionUnq`. Otherwise, the first Poll for the current date that the user has not already taken will be displayed.

- Enter all the code (HTML, ASP, JavaScript, etc.) that you need before the poll.

- If you are specifying which Poll to display, set the `iQuestionUnq` variable to the ID number of the poll you wish to display: `iQuestionUnq = 11` You can find the Poll ID number on the main poll administrative screen to the left of the polls question in the "ID" column.

- Call the function to display the `SB_POLL` Action: `Call DISP_Action("SB_POLL")`

- Enter any additional code (HTML, ASP, JavaScript, etc.) that you need after the poll.

Other Poll Pages

Poll Configuration Variables

- `SB_PUBLIC_POLL_TABLEWIDTH` – default: 671 – This is the width of the Poll Results and Past Polls pages.

- PUBLIC_COLORS – default: 3 (Aqua) – This is the Color Scheme of the Public System - the Poll pages are in this System.

- PUBLIC_TEXT – default: 1 (English) – This is the Text Scheme of the Public System - the Poll pages are in this System.

Public Poll Pages Actions

The pages that display poll results and past polls both use the same opening and closing Actions. You can customize them to match your own opening and closing HTML and ASP code:

- Go to the /WS_Content/ directory, Domains System, and Actions page to edit these Actions. Change the System drop-down list to the "For publicly viewable pages in SiteBeater" System (System #5).
  - Change the SB_PUBLIC_OPEN Action to change the opening code.
  - Change the SB_PUBLIC_CLOSE Action to change the closing code.
  - Change the SB_PUBLIC_CSS Action to change the StyleSheet code - if you change the StyleSheets you may have to also change the SB_PUBLIC_OPEN and SB_PUBLIC_CLOSE Actions to match your new style variables.

Poll Cookies

Users can only vote for a poll once until they close their browser. Once they close their browser, they can re-vote for the poll.

If a user has already voted for a poll, the system will display the next poll (based on poll id number) whose start and end date encompass the current date/time.

If you would like the cookie to expire at a later date (not expire when they close their browser), you can edit the /WS_Content/Public/Polls/Results.asp page and uncomment the line: `Response.Cookies("POLL" & iDomainName).Expires = CDate(DateAdd("d", 365, NOW))` on line 76 and changing the 365 to the number of days you would like the cookie to last.

Domains

The SiteBeater Foundation

10

Actions

Summary:	Use Actions to build websites and functionality.
Audience:	Administrator, Developer
Before You Begin:	Design each website.
Topics Include:	Benefits of Actions Using Actions Coding Actions Colors & Text Coding Actions Into Pages Action Ideas
See Also:	Creating Custom Systems, SiteBeater API

The purpose of SiteBeater is to create websites; Actions could be considered the bricks and mortar holding these websites together. Sometimes Actions are referred to as "macros". They can be called from your ASP code, or from other Actions. They can contain any HTML, text, JavaScript, Java calls, Flash calls, or even ASP. You can also use Text and Color Scheme data within Actions to make your website multilingual and multi-themed.

Benefits of Actions

When the code in an Action is changed, the change appears in all places the Action is called from. This is great when things like navigation are the same across many pages. Make the change in the Action, and the change is reflected throughout your website.

Another benefit is that the Actions are in a web-based interface. If a bug is found on a website at two in the morning, the developer does not have to drive to the office to fix it. She can just hop online and make her changes.

Security is an additional benefit. Many companies either have their web pages on a network, or at a hosting service, or both. Employees can mistakenly delete files, and hackers can sneak in to delete or change your files. Files are also more susceptible to virus attack (such as Code Red). With the data in a database in the form of Actions, there is less chance of virus attack or hacking. Also, hackers must break through Microsoft's MSSQL login and password system to break into the database to change the data.

Using Actions

Developers can create, view, delete, and modify Actions by:

- going to any SiteBeater enabled website on which you have rights to manage Actions
    - `SB_MACT_CREATE_NEW` to create new Actions
    - `SB_MACT_UPDATE` to edit existing Actions
    - `SB_MACT_DELETE` to remove Actions
    - `SB_MACT_USE_ASP` to use ASP in Actions and to be allowed to edit Actions with ASP in them. Actions are determined to contain ASP if they have the value of the `cCODE_DELIMITER` Configuration Variable within the Actions text.
- click on the Domains link in the top navigation
- click on the Actions link in the navigation beneath that
- from here, choose the Domain from which to administer Actions
- also choose the System from which to administer Actions

SiteBeater allows the use of either a plain text editor, or the Edit-On Pro WYSIWYG editor. To use the WYSIWYG editor:

- Make sure the editor is installed according to the editors documentation.
- Set the `USE_EDIT-ON` Configuration Variable to "YES"
- The WYSIWYG editor is best suited for Actions consisting of purely browser-based technologies like HTML and JavaScript. ASP displays poorly in the editor.
- The editor is best when the Action has completely self-contained HTML. An Action that is not self-contained would be an Action that opens an HTML table, but relies on a second Action to close it.

A basic websites scenario uses Actions to open and close web pages, and display navigation. For example, the SiteBeater administrative pages use an Action to open each page, and one to close each page. The Action that opens each page calls another Action to output the CSS StyleSheet code for the page.

SiteBeater Actions are broken up into various Systems. We suggest creating a new System specifically for custom Actions. Your website may have several opening and closing Actions. Create these in the new System to keep things organized.

Coding Actions

Once an Action has been created, it needs to have code or text added to it. An Action can contain any text, including HTML, JavaScript, Flash or Java calls and ASP. Text and HTML in an Action is just displayed. ASP within an Action is actually parsed and run. For a developer to add ASP to an Action, they must know the `cCODE_DELIMITER` Configuration Variables value. The default `cCODE_DELIMITER` value is `%PW%`. The `cCODE_DELIMITER` value must appear at the beginning and end of any block of ASP code within an Action.

For example, to print out the value of a variable called `sValue`, follow the format described above by using this text:
`%PW%Response.Write sValue%PW%.` Developers who have used ASP know that they can use shortcuts like `<%=sValue%>` to print out values. This is not allowed in Actions. Using the `cCODE_DELIMITER` is the only way to use ASP in an Action.

Another example:

```
%PW%
If ( bHasAccount = TRUE ) Then
      Response.Write "<a href='/Logout.asp'>Logout</a>"
Else
      Response.Write "<a href='/Login.asp'>Login</a>"
End If
%PW%
```

ASP variables, subroutines and functions may also be used within Actions. These have the same <u>scope</u> as they would if the `DISP_Action()` (the API function that runs Action code) call were replaced with the Actions code itself in the calling file.

For example:
Create an Action called “SmallAction”. In it, add the code:

```
      %PW%x = x + 1%PW%
```

In your ASP page, add the HTML and ASP:

```
<%x =1%>
<table width=100%>
      <tr>
            <td>
                  <%Call DISP_Action(“SmallAction”)%>
                  <%=x%>
            </td>
      </tr>
</table>
```

This will print out a 2 in the table. This is because the x in the webpage is the same x in the SmallAction Action – the scope is as if the Action was replaced by the Actions *actual* ASP code. The following code is how the computer *sees* what we have done:

```
<%x =1%>
<table width=100%>
     <tr>
          <td>
               <%x = x + 1%>
               <%=x%>
          </td>
     </tr>
</table>
```

The call to the Action is transparent to the computer.

Important: developers cannot break up code blocks into separate %PW% blocks. For example, if you have an if/then statement, you cannot have the "If (1 = 2) Then" in one %PW% block, and the "End If" in another %PW% block, both must be in the same %PW% block. This is because SiteBeater executes each %PW% block independent of each other block.

Beware of changing the `cCODE_DELIMITER` Configuration Variable. If the `cCODE_DELIMITER` is changed, all Actions that use its value must also be changed or they will not run the ASP. Changing the delimiter means that all SiteBeater administrative pages will loose their navigation because the Actions on those pages use ASP. Be careful!

Protect Actions that use ASP by limiting the Domain System `SB_MACT_USE_ASP` rights. Developers without this right cannot add ASP to any Action, even if they know the value of the `cCODE_DELIMITER` Configuration Variable. Also, developers without this right cannot edit Actions that already have ASP in them. The `cCODE_DELIMITER` is also protected by the `SB_MCONG_EDIT_cCODE_DELIMITER` (to edit it) and `SB_MCONG_VIEW_cCODE_DELIMITER` (to view it) rights. Only administrators with those rights can view or edit that variable.

Colors & Text

Actions can use Color and Text Scheme data. This allows developers the ability to create themes and languages for their website and use that information in their Actions.

On the Action editing page, there are two links – one for colors and one for text – that pop open new windows with either color or text data in them that can be added to the Action. Choose color and text variables from these windows to put into the Action code. Or, type the codes in by hand.

The codes entered are in the form of:

- “[`SBColor` *`colorname`*]” for colors
- “[`SBText` *`textname`*]” for text

For example:
`<font color=’[SBColor MainPageColor]’>[SBText ProductDetail]</font>`

The color or text name is the color or text constant associated with the entry in the scheme. If desired, developers can call the actual color or text display functions (see the sections on Text and Color Schemes). The “[]” characters are used rather than “<>” because, the color and text codes are often within other tags (such as <font> tags) and the WYSIWYG editor does not allow for embedded HTML tags.

Once an Action is created and code added to it, it must be added to your website.

Coding Actions Into Pages

Developers should use the API function `DISP_Action()` to display Action text and HTML. ASP within Actions will be parsed and run instead of displayed. See the section on Creating WebPages & Custom Systems for details on what should go in each ASP file before displaying your Actions.

A simple example of using Actions in a webpage:

```
<body>
<table width=100%>
     <tr>
          <td colspan=2>
               <%Call DISP_Action(“OpeningNav”)%>
          </td>
     </tr>
     <tr>
          <td width=100>
               <%Call DISP_Action(“SideNavigation”)%>
          </td>
          <td>
               This is the body of the page
          </td>
     </tr>
     <tr>
          <td colspan=2>
               <%Call DISP_Action(“ClosingNav”)%>
          </td>
     </tr>
</table>
</body>
```

This example does not include the opening and closing code needed in each page to make the page SiteBeater enabled. See the section on Creating WebPages & Custom Systems for examples of that code.

Actions can be embedded within Actions. For example, a Stylesheet Action could be called within the Action that opens each webpage. Or, a Java applet Action could be called within the body of a page Action.

To see which Actions are being used where, go to the page in question and view the source HTML. Each Action opens with text of this format:

```
<!----------- SiteBeater Version 2.00 - Begin: SB_STANDARD_OPEN ------------->
```

The comment includes the version of SiteBeater and the Action name. Actions close with similar text and include the Server ID number:

```
<!----------- SiteBeater END: SB_STANDARD_CLOSE : 3 ------------------------>
```

Action Ideas

Almost anything can be coded into Actions. Most of the time they are simply used for creating the structure of the sites pages. But, do not forget that ASP can be added to them.

- Many times navigation has links that change depending on if the user is logged in or not, or if the user has a certain right or not. Developers need to take advantage of this in the Actions they code. For example, change the login link to a logout link when a user is logged in. Add links for members that only show up when a member is logged in. Create a link just for administrators to access private pages.

- One of the best uses of the WYSIWYG editor is to edit the content of web pages. Actions are not limited to navigation. Some of the best websites use Actions to display the body of pages. It is a quick and simple way of editing content.

- Create an Action repository. Set up several different kinds of Actions – news or weather feed Actions, an Action to display user information, an Action to display stock prices or inventory levels or products in a store. It's easy to pass data to the Action telling it which product to display in an online store.

11

Colors & Color Schemes

Summary:	Color Schemes give websites the ability to change its looks on the fly.
Audience:	Administrator, Developer
Before You Begin:	Develop a color scheme and look for each website.
Topics Include:	Using Color Schemes Color Schemes & Configuration Variables Color Scheme Example
See Also:	Actions, Configuration Variables, Creating Custom Systems, SiteBeater API

Color constants and Color Schemes are what allow websites and Systems created using SiteBeater to change their looks quickly. The administrator creates Color Schemes and adds color constants to it. Each compatible scheme has the same constants in it. Each web page uses those constants in places that affect colors and looks. The color constants return the values that the administrator put in them previously and return those color values when the web page is displayed. When the scheme is changed, the constants return values from the new scheme. This has the effect of changing the look of the site.

Stylesheet information, image names, color tags, and more can all be placed into the value field of a color constant. Any text up to 255 characters may be input into each color constant.

Colors and color schemes are copied when you create a new SiteBeater Domain and use an existing Domain as its source. Once the colors and Color Schemes have been copied, they are completely independent of the source SiteBeater Domain and can be changed or removed, as necessary.

The SiteBeater Foundation includes three default Color Schemes. Each additional SiteBeater System (Message Board, Image Gallery, etc.) includes color constants for those three Color Schemes.

To manage Color Schemes, go to:

- any SiteBeater enabled website on which you have rights to manage Color Schemes:
    - `SB_MC_NEW_COLOR` – create a new color constant in the scheme
    - `SB_MC_REMOVE_COLOR` – remove color constants from schemes
    - `SB_MC_UPDATE_COLOR` – change color constant names and values
    - `SB_MC_NEW_SCHEME` – create new Color Schemes
    - `SB_MC_DELETE_SCHEME` – delete existing Color Schemes and their color constants
- click the Domains link in the top navigation
- click the Colors link in the navigation beneath that
- choose the Domain to manage
- choose the Color Scheme to manage

Using Color Schemes

Creating a New Scheme:

- One of the best methods of creating a new Color Scheme is by using an existing Scheme as a template.

- With the new copy of the scheme, the administrator can replace the old color values with the new values for the new scheme. For example, let us say that the original scheme included many shades of green, and that the new scheme will include many shades of red. The administrator will need to go through each color constant and change it's value from a green hue (say "11FF55") to a red hue: "FF1155".

- New web pages or systems will also require new color constants. The administrator can simply create the new constants by adding them from the Color Scheme management pages.

- Then, she can use her new color constants in her:
    - web pages ASP – using the `DOMAIN_Color(`*`color constant name`*`)` API function.
    - or Actions – using the Action format for adding color constants: `[SBColor `*`color constant name`*`].`

- Administrators may need multiple variations of a custom Color Scheme. After creating the first version of the new Color Scheme, the administrator can then create the other variations of their custom Color Scheme. The administrator would use this first scheme as the source scheme. Each new scheme would already have the color constants, and only their values would need to be changed.

Debugging Color Schemes:

- If a web page is requesting a color constant that is not in the current Color Scheme, the constants name will be returned rather than any value. This usually results in strange colors on the web page, but allows the developer to see which color constants are missing.

Color Schemes & Configuration Variables

Each SiteBeater System has a Configuration Variable associated with its Color Scheme. This Configuration Variable is what actually specifies which scheme to use within that System. For example, the Accounts System uses the `ACCOUNTS_COLORS` Configuration Variable to tell it which Color Scheme to use. When this Configuration Variable is changed on the Configuration Variables management screen, the colors in the Accounts System change. If this variable is set to the “Aqua” scheme, the colors are green. If it’s set to the “White” scheme, the colors are white and gray.

When a custom system or web pages are created, a custom Configuration Variable can also be created to specify which Color Scheme to use for the system or pages.

For example:

- Let us say that your developers are creating a new website. This website will use a completely custom set of Color Schemes. It does not make sense to use any of the existing SiteBeater scheme Configuration Variables. So, the developers create a new variable and call it `WEBSITE_COLORS`.

- When they need to change the scheme on their new website, they change the `WEBSITE_COLORS` Configuration Variable to whichever scheme they want.

- The web pages in the new website must have some way of knowing which Configuration Variable to use. SiteBeater has a global variable called `iColorScheme` that lets the SiteBeater API color function (`DOMAIN_Color()`) know which scheme to use. This global variable must be changed to the color scheme in each web page for the web page to know which color scheme to use. Remember from above that the color scheme is saved in the `WEBSITE_COLORS` Configuration Variable. Use the SiteBeater API function `DOMAIN_Conf(“WEBSITE_COLORS”)` to return the value of that Configuration Variable. At the beginning of every web page, the Color Scheme must be set, as such:
  ```
  iColorScheme = DOMAIN_Conf(“WEBSITE_COLORS”)
  ```

- The Configuration Variable uses the Color Schemes scheme number. The Configuration Variable management screen allows administrators to choose the Color Scheme from a list of schemes. Administrators do not usually need to know the scheme number. On the other hand, developers may need to know it in case they need to hardcode the numbers in their code. The scheme number can be found next to the scheme name in the Scheme drop-down list on the scheme-editing page.

A second example:

- Create a `WEBSITE_COLORS` Configuration Variable and custom Purple Color Scheme.
  - Set `WEBSITE_COLORS` to the custom Purple Color Scheme.
  - In the opening website Action, set the `iColorScheme` variable to the `WEBSITE_COLORS` Configuration Variable value: `%PW%iColorScheme = DOMAIN_Conf("WEBSITE_COLORS")%PW%`. The website will appear purple.

- Go to the login page. If the Purple Color Scheme has in it custom color constants that the login page does not use (remember that the login page is looking for the standard Accounts color constants), it will display the missing (they are in the Accounts Color Scheme, not in the Purple Color Scheme) color constant names in place of their values. Solutions to this problem include:
  - The color constants used on the login page either need to be added to the Purple Color Scheme.
  - Or, the `iColorScheme` needs to be set back to the Accounts Color Scheme.
  - Or, as a third alternative, a new Default Purple Color Scheme could be created that used one of the default SiteBeater Color Schemes (say Aqua). Then change its color constants values to match the Purple Color Scheme – it will then have all the color constants that the login page is looking for.

- Developers must write code to match the Color Scheme they plan on using. For example, the Purple Color Scheme may contain a color constant called `HEADER_TEXT`. There exists another custom scheme called Red Color Scheme does not have the `HEADER_TEXT` color constant. If the administrator switches from the Purple to Red Color Scheme, any place in the code that uses the `HEADER_TEXT` color constant will display incorrectly. This is because the code was looking in the Red Color Scheme for a color constant that the Red Scheme did not have.

12

Configuration Variables

Summary:	Set up SiteBeater Configuration Variables or create new variables for custom systems.
Audience:	Administrator, Developer
Before You Begin:	Gather information to populate the variables.
Topics Include:	List of Configuration Variables
See Also:	Creating Custom Systems, Setting Server IP Addresses

Configuration Variables hold information that the SiteBeater System uses. This information ranges from passwords to Color Scheme numbers. Administrators can change the Configuration Variables that come with the different SiteBeater Systems.

Manage Configuration Variables by going to:

- any SiteBeater enabled website on which you have rights to manage Configuration Variables:
  - `SB_MCONF_CREATE_NEW` – create new variable
  - `SB_MCONF_DELETE` – remove existing variable
  - `SB_MCONF_UPDATE` – change existing variable
  - `SB_MCONG_EDIT_RIGHTS` – change the rights needed to update variables
- go to the WS_Content directory
- click the Domains link in the top navigation
- click the Configurations link in the navigation beneath that

Administrators should carefully track who has access to view or change Configuration Variables. If a user has right to update, add, or delete Configuration Variables on a SiteBeater Domain, they may also be able to see private ones on that Domain. Some of the variables hold sensitive information like passwords.

And, an administrator can add more critical or private custom variables (merchant account number, access passwords, etc.).

Configuration Variable Rights

In addition to the rights to add, edit and delete Configuration Variables, an administrator may have the right to change Configuration Variable rights (`SB_MCONG_EDIT_RIGHTS`). Each Configuration Variable may be assigned a right level that is required for an administrator to either view or change it. This right allows site owners to give users the right to change select Configuration Variables.

For example: a site owner may want to delegate some authority to a user and allow them to change the color Configuration Variable on the site. The administrator would not want that user to be able to see (much less edit) the `DB_ACCOUNTS_PASSWORD` Configuration Variable (the password to the SB_Accounts database). Before giving the user rights to edit Configuration Variables (`SB_MCONF_UPDATE`), the administrator should set the view rights on `DB_ACCOUNTS_PASSWORD` to a rights level that the user does not have. The administrator could require that a user have rights to view live DB connection information (`SB_CONF_VIEW_LIVE_DB`) to see the `DB_ACCOUNTS_PASSWORD` Configuration Variable. The administrator could also create a custom right, say `VIEW_DBs` and assign that to the `DB_ACCOUNTS_PASSWORD` Configuration Variable.

Using Configuration Variables

The interpretation of each Configuration Variable reflects the way the SiteBeater System is currently coded. If the source code is changed to interpret the variables differently, then the variables can be used in other ways. For example, the `ACCOUNTS_SIGNUP` looks for either “OPEN” or “CLOSED”; meaning signing up for new memberships is either open or closed to the public. Developers could add a third possibility: “PAY ONLY” to mean users need to purchase a membership. Then they would have to change the ASP in the web pages to be able to handle the new “PAY ONLY” status of the variable.

Use the `DOMAIN_Conf(“sVariableName”)` API function to request the value of a Configuration Variable. See the API documentation for more information.

Configuration Variables may be associated with an account Group. This is useful when a specific group of users need different site configurations. For example, let us say that a company would like all users in a Gallery Editors Group to have unlimited hard disk space for adding images to galleries. The administrator would create a new copy of the `IMAGEGALLERY_INITIAL_NUMDOWNLOADS` Configuration Variable. This copy would be assigned to the Gallery Editors Group and be given the value of –1 (meaning unlimited). Now all accounts within the Group will be allowed to upload an unlimited number of bytes to the Image

Gallery. (Unfortunately this Configuration Variable only affects new accounts, so all existing accounts would have to be updated by hand – see the Image Gallery documentation for more information on this Configuration Variable.)

List of Configuration Variables

Foundation Variables

Variable	Default Value	Information
`SB_ACCOUNTS_SERVER`	*the value entered at the time of installation*	This is the name of the database server machine that the SB_Accounts database is on. The LIVE server version requires `SB_CONF_EDIT_LIVE_DB` rights to edit and `SB_CONF_VIEW_LIVE_DB` rights to view.
`SB_ACCOUNTS_LOGIN`	SB_Accounts	This is the login for the SB_Accounts database. The LIVE server version requires `SB_CONF_EDIT_LIVE_DB` rights to edit and `SB_CONF_VIEW_LIVE_DB` rights to view.
`SB_ACCOUNTS_PASSWORD`	*the value entered at the time of installation*	This is the password for the SB_Accounts DB. The LIVE server version requires `SB_CONF_EDIT_LIVE_DB` rights to edit and `SB_CONF_VIEW_LIVE_DB` rights to view.
`cCODE_DELIMITER`	%PW%	This is the text that opens and closes ASP code in SiteBeater Actions. See the section on Actions. Changing this Configuration Variable does NOT change code within Actions to the new cCODE_DELIMITER. Requires `SB_CONF_EDIT_cCODE_DELIMITER` rights to edit and `SB_CONF_VIEW_cCODE_DELIMITER` rights to view.
`ACCOUNTS_COLORS`	3 (Aqua)	This is the SchemeUnq of the color scheme to use on the Accounts System. This includes the login, page to turn off auto-login, signup, and preferences pages. See the section on Color Schemes. Requires `SB_CONF_CHANGE_ACCOUNTS_COLORS` rights to edit and `SB_CONF_VIEW_ACCOUNTS_COLORS` rights to view.
`DOMAINS_COLORS`	3 (Aqua)	This is the SchemeUnq of the color scheme to use on the Domains System. See the section on Color Schemes. Requires `SB_CONF_CHANGE_DOMAINS_COLORS` rights to edit and `SB_CONF_VIEW_DOMAINS_COLORS` rights to view.
`ACCOUNTS_TEXT`	1 (English)	This is the SchemeUnq of the text scheme to use on the Accounts System. See the section on Text and Text Schemes. Requires `SB_CONF_CHANGE_ACCOUNTS_TEXT` rights to edit and `SB_CONF_VIEW_ACCOUNTS_TEXT` rights to view.
`DOMAINS_TEXT`	1 (English)	This is the SchemeUnq of the text scheme to use on the Domains System. See the section on Text and Text Schemes. Requires `SB_CONF_CHANGE_DOMAINS_TEXT` rights to edit and `SB_CONF_VIEW_DOMAINS_TEXT` rights to view.
`ACCOUNTS_SIGNUP`	OPEN	If this is set as "OPEN", then anyone can signup for a new account. If it's set as "CLOSED", then administrators must create them in the Manage Accounts screen.
`EMAIL_TYPE`	ASPMAIL	Either “ASPMAIL” or “CDONTS”. This is the

		method or program used to send emails. Set it to "ASPMAIL" to use ASPMail or ASPQMail from ServerObjects Inc. (www.serverobjects.com).
EMAIL_LOGIN_FROMNAME	Customer Support	Used when a user is retrieving their login/password. This is the name of the sender.
EMAIL_LOGIN_FROMEMAIL	Customersupport @ YourWebSiteHere .com	Also used when a user is retrieving their login/password. This is the email address of the sender.
EMAIL_REMOTEHOST	mail.YourWebSiteHere.com	Email server used to route messages. This Configuration Variable comes with the SiteBeater Foundation and is used when sending emails from the different systems. These hold the name of the remote host used to send emails through (usually in the form of "mail.domainname.com"). The ASPMail documentation (www.ServerObjects.com) has more information on this variable. Clear this value to turn off all emailing abilities on the website.
USE_EDIT-ON	NO	Set to "YES" to use the Edit-On WYSIWYG editor. "NO" to not use it, or if it is not installed.
EDIT-ON_LICENSEKEY		The Edit-On Pro license key for this SiteBeater Domain. Populate this only when not using the standard licensekey.xml file.
EDIT-ON_SPELLCHECKPROPERTIES	sc-americanenglish.txt	Set to the spell check properties file to use. Change according to the language. See the Edit-On Pro documentation.
EDIT-ON_IMAGEROOT	http://www. YourWebSiteHere.com/ Images	ImageRoot value from Edit-On Pro. See the Edit-On Pro documentation. With multiple web servers and one of them being reserved as an image server, this can be used to point to that server
EDIT-ON_TEXT	1 (English)	The Text SchemeUnq used with Edit-On. Must be a SiteBeater Text Scheme or scheme created using a SiteBeater Text Scheme, as it needs text constants only found in default SiteBeater Text Schemes. See the section on Text Schemes for more information.
EDIT-ON_IMAGEPROXYURL		Use with the EDIT-ON_IMAGEROOT. The IP address/Domain name (with out the http://) of the server hosting all images (e.g. 192.168.1.1) – or leave blank if there is no specific image server.
ACCOUNTS_TABLEWIDTH	671	Width of the tables in the login.asp, disableautologin.asp, new user signup page, exiting user preferences page, and the i_Domains.asp function DOMAIN_Message(). Set this to a pixel number, or percent of page width.
SEND_ADMIN_REG_EMAIL	YES	Set to "YES" to send an email to the administrator when a user creates a new account, or "NO" to not.
SEND_ADMIN_REG_EMAIL_DEST	administrator@ YourWebSiteHere.com	For use with SEND_ADMIN_REG_EMAIL. Set to the administrators email address.
SEND_ADMIN_REG_FROMNAME	SiteBeater Administrator	For use with SEND_ADMIN_REG_EMAIL. Set to the name of the person who sends the email.
SEND_ADMIN_REG_FROMEMAIL	newuser@ YourWebSiteHere .com	For use with SEND_ADMIN_REG_EMAIL. Set to the email address of the person who sends the email.
SEND_ADMIN_REG_DESTNAME	Administrator	For use with SEND_ADMIN_REG_EMAIL. Set to the name of the administrator who receives the email.
LOGGING_ENABLED	Enabled	Enter anything in this variable to enable logging. Clear this to disable logging.
SB_PUBLIC_POLL_TABLEWIDTH	671	This is the width of the tables on the results pages of the public Poll pages. This can be either a number of pixels or screen width percent.
PUBLIC_COLORS	3 (Aqua)	This is the SchemeUnq of the color scheme to use on the Public system.

`PUBLIC_TEXT`	1 (English)	This is the SchemeUnq of the text scheme to use on the Public system.

13

SiteBeater Domains

Summary:	SiteBeater Domains link all the data to a website.
Audience:	Administrator, Developer
Before You Begin:	Decide who will modify the SiteBeater Domains.
Topics Include:	Creating a New SiteBeater Domain Deleting Domains SiteBeater Domain Security Domain Groups
See Also:	Rights, Foundation Setup

Internet domains reference SiteBeater Domains. The SiteBeater Domain is simply a way of assigning data (such as Text and Color Schemes and Actions) to websites. Each of your websites must have a SiteBeater Domain entry. Until your website has a SiteBeater Domain entry, it will not be able to use any SiteBeater functionality or Systems.

When a SiteBeater Domain is changed, its corresponding IIS entry may also be changed.

Manage Domains by going to

- Any SiteBeater enabled website on which you have rights to manage Domains:
    - `SB_MD_CREATE_NEW` – right to create new SiteBeater Domains or websites
    - `SB_MD_DELETE` – rights to delete existing Domains
    - `SB_MD_UPDATE` – rights to change Domain name
    - `SB_MD_MODIFY_DOMAIN` – rights to modify Domain Groups
- Go to the WS_Content directory
- Click on the Domains link in the top navigation
- Click on the Domains link in the navigation beneath that

Creating a New SiteBeater Domain

After the initial SiteBeater Foundation installation, Server information must be entered before creating a new SiteBeater Domain. The installation script cannot enter complete Server information. The Foundation installation script will create three server entries based on information gathered from the server used to perform the installation. This information does not include several paths, or the administrator login or password for the servers. Before creating any new SiteBeater websites, please enter the missing path and login information. Go to

- Any SiteBeater enabled website on which you have rights to manage Servers:
  - SB_MSE_CREATE_NEW – create new Server entries
  - SB_MSE_UPDATE – update existing Server entries
  - SB_MSE_DELETE – delete Servers
- Go to the WS_Content directory
- Click on the Domains link in the top navigation
- Click on the Servers link in the navigation beneath that (see the section on Multiple Servers for more information about Server).

Creating the New Domain

When creating a new SiteBeater Domain, several options are present and requirements for those options:

- **Using a Source Domain.** An existing SiteBeater Domain may be selected to copy the SiteBeater data from. This copies all Text and Color Schemes, Configuration Variables, ADVs, Initial Rights, Actions, and Rights (but not necessarily rights assignments – see the next item).

- **Copying Rights from Source Domain.** Independent of using a Source Domain, rights assignments may also be copied to the new domain. Copying the rights means that users with rights to do something on the source SiteBeater Domain will have those same rights on the destination Domain. This also copies all Groups and Group rights.

- **Grouping the new Domain.** The administrator may also choose to group the new Domain with an existing Domain. The section on Domain Groups (below) has more information on Domain Groups. Adding the new Domain to a Domain Group does not give any existing users the right to log onto the new Domain. It only affects new users who create their account on a Domain in that Group.

- **Enabling users to log onto new Domain.** If the administrator chooses to add the new Domain to the list of Domains users can log onto, then existing users may be able to log onto the new Domain. This only affects users who have Home Domain(s) that the administrator has rights on to give users login rights. In other words, for a user to be given the right to log onto the new Domain, they must have a Home Domain that the administrator has `SB_MA_CHANGE_DOMAINS` rights on.

- **Copying files from Source Domain.** The administrator may also choose to have all of the files from the source Domain copied to the destination Domains directory. This is great when making many copies of the same website, or a template website.
    - If the administrator chooses to copy files, she must enter the source directory. The source directory does not need to match any existing Domains directory (i.e. it is not required to be the Source Domains directory, or the Grouping Domain).

- **Copying files – part II, and/or Creating an IIS Entry.** To create either a new directory or any IIS entries for the new Domain, the administrator must enter the new website directory name. This is used for both copying files and creating an IIS entry. IIS uses this information to know where the website is stored on the server. It is also the destination directory created for the new Domain when the administrator chooses to create the new directory for the new Domain. Enter it if one or the other, or both choices are made.
    - After copying files, the directory and file permissions must be set correctly. For example, if the source SiteBeater Domain has a directory within it for images in the Image Gallery System, the administrator must make sure the same directory in the destination has the same (IUSR modify) rights. If it does not, no one will be able to upload images on the new Domain. The exception to this is if the administrator has set the rights to the Domains directory to cascade to all subdirectories.

- **Creating an IIS Entry – part II.** SiteBeater can create the IIS entries when making the new Domain in SiteBeater.
    - The administrator must enter the new websites directory name even if they do not intend to have SiteBeater create the directory.
    - The administrator must also enter the Server path (this is found on the Server management page – Domains->Server). The new directory name and Server path are what make up the complete path to the new Domain in IIS.
    - If the administrator chooses to not have SiteBeater copy source files to the new Domain, a new directory must be created by hand and the files copied to it.
    - See the section on Multiple Servers and Security Issues for information about the requirements for creating Domains and IIS entries on multiple servers.
    - If SiteBeater does not create the IIS entry for the new Domain, it cannot manage it in IIS. This is usually only a minor inconvenience. When there is a change to the Domain name or the

Domain is removed, the administrator must make parallel changes to the Domain in IIS because SiteBeater will not make the changes for them. For example, if the administrator changes a Domain name in SiteBeater, she must also change the same domain in IIS.

- **Not Creating IIS Entries.** If SiteBeater did not create IIS entries, a new Virtual Directory must be created for the new Domain in IIS. This Virtual Directory must point to SiteBeaters /WS_Content/ directory.
  - The Virtual Directory does not to be created if the WS_Content directory has already been copied to the new Domains directory.

It is possible to work on a website before purchasing the Internet domain. A *Microsoft Loopback Adapter* or self-referencing DNS and IIS servers will allow developers to still work on the site. This allows for development even before purchasing the domain.

This error may appear on new websites: "*Critical Error: Domain name not found in database. Quitting.*" It generally appears when a new SiteBeater Domain has not been created for the Internet domain.

This error could also occur when accessing a website using its IP address – http://123.123.123.123 instead of domain name – http://www.mysite.com. If, when creating the SiteBeater Domain, the administrator used "www.mysite.com" as the Domain name, the users must type that exact text (case insensitive) in the browsers address bar. To use the IP address, change the SiteBeater Domains name to the IP address numbers ("123.123.123.003").

Using an IP Address Instead of Internet Domain Name
Normally the administrator would use the Internet domain name as the new SiteBeater Domain name. Another option is where websites are accessed via their IP addresses. This happens most during testing or before purchasing the Internet domain name. To access a website by its IP address, enter the IP address as the Domain name when creating the new SiteBeater Domain, or when running the SiteBeater Foundation installation script for the first time. The name of the Domain can be changed to a standard URL later on by changing it in the Domains management screen.

Using *localhost* as the URL of the Domain is not supported.

Deleting Domains

To delete any part of a Domain, an administrator must have `SB_MD_DELETE` rights. An administrator may delete the Domains database data, files, or IIS entries.

When a Domains database data is deleted, it removes everything, including:

- Configuration Variables
- Color and Text Schemes
- Actions
- Servers it was on
- Rights
- Initial Rights
- Groups
- Mailing Lists
- and disassociates users who have it as their Home Domain.

Log entries are not created for the deletion of these items when a Domain is deleted.

There are several ways to delete a Domain and parts of it that can be deleted, changed or shut off. These actions often require special setup before being able to perform them:

- **Delete a Domain from IIS on the current server**. For SiteBeater to manipulate IIS:
  - TqcRunas must be installed
  - the Domain must have been created on the server using SiteBeater
  - NT Authentication must be enabled on the website you are changing IIS from
  - You must be logged into the NT Authentication as an administrator with NT rights to manipulate the metabase (to change IIS).

- **Delete a Domains directory on the current server.**
  - The directory must have permissions set so that the IUSR can delete it.

- **Delete the Domains directories or IIS entries from multiple servers.** This also applies to a server other than the one that serving the current SiteBeater page:
  - TqcRunas must be installed
  - the Domain must have been created on those servers using SiteBeater
  - NT Authentication must be enabled on the website you are deleting the Domain from
  - You must be logged into the NT Authentication as an administrator with:
    - NT rights to delete files and directories on all of the servers
    - NT rights to manipulate the metabase (to change IIS) on all of the servers.

- **Stop the website in IIS from the current, multiple or other servers**:
  - TqcRunas must be installed

  - the Domain must have been created on all servers using SiteBeater
  - NT Authentication must be enabled on the website you are deleting the Domain from
  - You must be logged into the NT Authentication as an administrator with:
    - NT rights to delete files and directories on all of the servers
    - NT rights to manipulate the metabase (to change IIS) on all of the servers.

- **Rename the websites directory on the current server.**
  - The directory must have permissions set so that the IUSR account can rename the directory.
  - The `DOMAINS_DELETE_DOMAINS_MOVED_FOLDER` text constant contains the value that is appended to the directory name when it is renamed.

- **Rename the websites directory on another server, or multiple servers.**
  - TqcRunas must be installed
  - the Domain must have been created on all servers using SiteBeater
  - NT Authentication must be enabled on the website you are deleting the Domain from
  - You must be logged into the NT Authentication as an administrator with:
    - NT rights to delete files and directories on all of the servers
    - NT rights to manipulate the metabase (to change IIS) on all of the servers.

- **Delete the Domains database data.**
  - No additional permissions or settings required.

When an administrator chooses to change a Domain on multiple servers, he can select the servers to change. The same server will appear several times if there are ALPHA, BETA, or LIVE versions of the server. The administrator only needs to choose one of the variations of the server for the Domain to be affected on that server. The variations are listed only for reference.

Domain Groups

Domain Groups are Domains that new users can log onto. If a user creates an account within any Domain in a Domain Group, that use can log onto any of the Domains in that Group. Changing Domain Groups only affects new users. **Change the Domains an existing user can log onto by going to the Accounts System Manage Accounts page.**

Administrators must have `SB_MA_CHANGE_DOMAINS` rights on the users Home Domain to change which Domains an existing account can log onto. They must have `SB_MD_MODIFY_DOMAIN` rights on a Domain in order to add it to a Domain Group.

SiteBeater Domain Security

Being able to modify SiteBeater Domains is a powerful right.

If an administrator creates a new Domain and copies the rights from a Source Domain, the new Domain will possibly have rights for accounts that cannot log into it. This is because those users do not have the new Domain in the list of Domains they can log onto.

Also, no account has the new Domain as one of the Domains they can log onto. So, when the Domain is added to the list of Domains an account can log onto, that account will appear to suddenly have rights on the new Domain. The account actually had those rights the whole time, it just could not log in, so could not tell.

Avoid this situation by

- grouping the new Domain with another Domain at creation time (from the Domain Grouping drop-down list)
- checking the box under that to add the new Domain to the list of Domains users (whom you have rights to) can log onto.

If the administrator has rights to add Domains to the users on the current Domain and use the current Domain as a Source Domain, then all the users covered will have access to the new Domain.

For information and requirements needed for administrators to use SiteBeater to administer IIS, please see the sections entitled Multiple Servers and Security Issues.

14

Systems

Summary:	Systems encapsulate code and data.
Audience:	Administrator, Developer
Before You Begin:	Decide what Systems to create and who has rights to manipulate them.
Topics Include:	System Implementations
See Also:	Domains, Actions, Rights, ADVs, Creating Custom Systems

What we call a "SiteBeater System" is simply an implementation of the SiteBeater API. Each System has a System Constant and ID number associated with it. The constant is for ease of use when programming – this is so developers do not need to remember each ID number. The ID number is what is used in the database to reference each System. Systems are also used to categorize Actions, Color and Text Schemes, Configuration Variables and other SiteBeater data.

Manage Systems by going to

- Any SiteBeater enabled website on which you have rights to manage Systems:
    - `SB_MS_CREATE_NEW` – right to create new SiteBeater Systems.
    - `SB_MS_DELETE` – rights to delete existing Systems
    - `SB_MS_UPDATE` – rights to change existing Systems
- Go to the WS_Content directory
- Click on the Domains link in the top navigation
- Click on the Systems link in the navigation beneath that

Administrators can create, edit, and remove System database entries. The System ID number is in several other database tables. The data in these tables reference the System using the System ID number. If a System ID number is removed or changed, the System ID numbers in these other tables do not change. Because the

System is gone, no administrative page can see any data from tables referencing the System ID number. This other data is essentially lost in the database. The System ID can be recreated again and the administrative pages will again be able to reference the data that is associated with that System ID.

For example, rights are all associated with a specific System. Right A is associated with System 12. If System 12 is deleted, the rights administrative pages cannot see System 12 or any rights associated with it – including right A. But, right A is still in the database, and still has 12 as its System ID number. If System 12 is recreated, the rights administrative pages will again be able to see it – as well as all rights within it, including right A.

Systems are SiteBeater Domain **independent**. This means that when one System is created on one SiteBeater Domain, it appears on all. If one Systems entry on one SiteBeater Domain is changed, that change appears and affects all Domains. This is so all Domains can share functionality. For example, if the SiteBeater Image Gallery System is installed, it can be used on any SiteBeater Domain.

System Implementations

Administrators are allowed to add Systems so they can organize Actions, rights, ADVs and custom code better. An example of a custom System is a website System that can be created to be used for all websites. Remember that a new Website System will be usable on all SiteBeater Domains. Developers can create all website Actions, colors, text, rights and Configuration Variables within that System.

Custom code should also be created under the umbrella of a new System. That way, it is easier to keep that code and data encapsulated and independent from the rest of the SiteBeater code, data and Systems.

Generally, custom code does not use the System constants directly. The System constants help the developer know which Actions, Configuration Variables, ADVs or rights are useable within a specific System. For example, the Image Gallery has several Actions associated with it. Those Actions were placed in the Image Gallery System (System Constant `IMAGEGALLERY_SYSTEMUNQ` which is number 7). A developer would then know immediately that he could use all Actions within System number 7 in the Image Gallery.

15

Text & Text Schemes

Summary:	Text Schemes give websites the ability to change languages on the fly.
Audience:	Administrator, Developer
Before You Begin:	Decide who will manage and translate each website.
Topics Include:	Emails Using Text Schemes Text Scheme Example

---

Text Schemes and constants are what allow websites and Systems created using SiteBeater to change their language on the fly. Text Schemes and constants work in much the same way as Color Schemes and Constants.

Manage Text Schemes by going to

- Any SiteBeater enabled website on which you have rights to manage Text Schemes:
  - `SB_MT_NEW_CONST` – right to create new text constants within schemes.
  - `SB_MT_REMOVE_CONST` – rights to remove constants from schemes
  - `SB_MT_UPDATE_CONST` – rights to change existing constants
  - `SB_MT_NEW_SCHEME` – right to create new Text Schemes.
  - `SB_MT_DELETE_SCHEME` – right to delete existing schemes and all associated constants.
- Go to the WS_Content directory
- Click on the Domains link in the top navigation
- Click on the Text link in the navigation beneath that

Administrators create Text Schemes and add text constants to them. Compatible schemes have constants with the same names in them (each scheme can, of course, have different constants). Each web page uses those constants in places that affect languages. When the site changes from scheme to scheme, the only thing that changes is the value of those constants. This has the effect of changing the language on the site.

Any information can be entered into the value field of each Text Constant. Often, images, Flash, or Java applets have text. Developers and administrators can use a text constant to hold the image or Flash file names for each different language. Then, put the Text Constant in their code instead of the image or Flash reference.

Sometimes a developer may need to add spaces before or after text in a text constant. SiteBeater automatically strips out all white space (spaces, tabs, crlf) when displaying text constants. Developers can add spaces by entering in the HTML &nbsp; tag. These tags will not be stripped out.

Text and Text Schemes are copied when a new SiteBeater Domain is created. Once the Text Schemes and Constants have been copied to the new Domain, they are independent of the source Domain and can be changed or removed as desired.

Administrators can use the Edit-On Pro editor to change long text values. Make sure the editor is installed and the `USE_EDIT-ON` Configuration Variable is set to “YES”. This editor is great because it has a built-in spell checker and can help with any HTML embedded in the text.

Emails

All of the SiteBeater emails are stored in text constants. Each email has text within it that is replaced when the email is sent. The replaced text is in the format “`X:`” where “`X`” is a number. Administrators can move these numbers around within the email as necessary. Some of the replaced text may be removed – some is required.

`EMAIL_RETRIEVELOGIN_BODY` – this is the body of the email that is sent to users who request a reminder of what their login and password is. This happens on the login page. This email contains `1:` and `2:`

- 1: - this is replaced with the users login
- 2: - this is replaced with the users password

`EMAIL_RETRIEVELOGIN_SUBJECT` – this is the subject of the email. Configuration Variables used with the reminder email include: `EMAIL_LOGIN_FROMNAME` and `EMAIL_LOGIN_FROMEMAIL`.

`NEWACCOUNTS_ADMIN_EMAIL_MSG` – this is the body of the optional email that is sent to the administrator when a user creates a new account. It contains `1:` and `2:`

- 1: - this is replaced with the new users login
- 2: - this is replaced with the new users password

`NEWACCOUNTS_ADMIN_SUBJECT` – this is the subject of the email.

Configuration Variables used: `SEND_ADMIN_REG_EMAIL`, `SEND_ADMIN_REG_EMAIL_DEST`, `SEND_ADMIN_REG_FROMNAME`, `SEND_ADMIN_REG_FROMEMAIL`, `SEND_ADMIN_REG_DESTNAME`.

`NEWACCOUNTS_USER_EMAIL_MSG` – this is the body of the email that is optionally sent to the user when they create a new account. It contains `1:` and `2:`

- 1: - this is replaced with the new users login
- 2: - this is replaced with the new users password

`NEWACCOUNTS_USER_EMAIL_SUBJECT` – this is the subject of the email. Configuration Variables used: `SEND_USER_REG_EMAIL`, `SEND_USER_REG_EMAIL_SOURCE`, `SEND_USER_REG_EMAIL_NAME`. The destination information comes from the new user information.

Using Text Schemes

The English Text Scheme is included with the SiteBeater Foundation. Other SiteBeater Systems insert additional constants into that scheme.

When creating a new Text Scheme, administrators may copy all of the text constants from a source scheme. This helps with translating. When a scheme is copied, all of the text constants are already in the new scheme. Translators only need to change the values. Consider giving translators rights to update (`SB_MT_UPDATE_CONST`) text constants.

The scheme number identifies each Text Scheme. The scheme number can be found next to the scheme name in the Scheme drop-down list on the scheme-editing page. The SiteBeater Systems use a Configuration Variable to save these values (see the section on Configuration Variables within each Systems documentation). For example, the `ACCOUNTS_TEXT` Configuration Variable holds the scheme number that is used on all of the web pages associated with the SiteBeater Accounts System. Administrators can create Configuration Variables to hold website Text Scheme numbers.

Text Scheme Example

Create a `WEBSITE_TEXT` Configuration Variable and a custom Chinese Text Scheme. In the website opening Action (you might use `SB_STANDARD_OPEN`), set the `iTextScheme` variable to the Configuration Variable's value. The website will appear in Chinese.

If a page used text constants not in the Chinese Text Scheme – say, for example the login page (remember the login page uses default text constants), it will display the missing text constants names in place of values.

Add the text constants that the login page is looking for to the custom Chinese Text Scheme, or set `iTextScheme` back to one of the default Text Schemes. As a third alternative, create a new Default Chinese Text Scheme from one of the default SiteBeater Text Schemes (English) and change the values of its text constants to Chinese.

Specialty sites might use Text Schemes that are variants on languages. For example, create a Jive, Pirate, Shakespeare, or Star Trek Text Scheme. Developers could even create a Text and Color Scheme combination to make a version of the website completely WAI Conformant for users with disabilities.

16

Migration

Summary:	Copy data from one website to another using the Migration tool.
Audience:	Administrator, Developer
Before You Begin:	
Topics Include:	
See Also:	SiteBeater Domains, Using Servers (ALPHA/BETA/LIVE), Multiple Servers

Migrating means copying the data (files and database data) from one Domain to another.

Migrate data by going to

- Any SiteBeater enabled website on which you have rights to Migrate data:
    - `SB_MIG_ACTIONS` – rights to migrate Actions.
    - `SB_MIG_COLORSCHEMES` – rights to migrate whole Color Schemes.
    - `SB_MIG_CONFVAR` – rights to migrate Configuration Variables.
    - `SB_MIG_TEXTSCHEMES` – right to migrate whole Text Schemes.
- Go to the WS_Content directory
- Click on the Domains link in the top navigation
- Click on the Migrate link in the navigation beneath that

SiteBeater allows administrators to copy data from one website to another using the Migration tool. Administrators can migrate Actions, Color or Text Schemes, and Configuration Variables. The migration overwrites any existing data.

When an administrator migrates data, she must have rights to migrate on both the source and destination domains. To migrate, the administrator must have rights to migrate the specific data type.

> Generally the migration tool is used to copy data from one Domain to another. Many times its easier to make changes on one website first, then rather than having to repeat the changes many times on other Domains, administrators can simply migrate the data.
>
> The migration tool can also be used to copy data from a custom System. New Systems are generally developed on a separate website. Once the System is finished, it’s new data (Actions, Color Schemes, etc) can be copied to any existing website. See the section on Creating WebPages & Custom Systems for more information.

Sometimes data is created in new Systems that cannot be migrated using the migration tool (such as new rights). This data must be copied to each website by hand or with SQL scripts.

17

Logging

Summary:	Logging tracks all changes administrators make.
Audience:	Administrator, Developer
Before You Begin:	Decide if logging should be used. Assign administrators to check logs.
Topics Include:	Custom Log Entries
See Also:	

SiteBeater logging tracks all changes made by administrators. Administrators may enable logging by changing the `LOGGING_ENABLED` Configuration Variable to “Enabled”. The `SB_CONF_CHANGE_LOGGING` right is required to change the `LOGGING_ENABLED` Configuration Variable. Any other value will disable logging.

Manage Logging and Logged data by going to

- Any SiteBeater enabled website on which you have rights to manage Logging:
    - `SB_LOG_VIEW_1` – rights to view Accounts System log entries.
    - `SB_LOG_VIEW_8` – rights to view Domains System log entries.
    - `SB_LOG_DELETE_1` – rights to delete Accounts System log entries.
    - `SB_LOG_DELETE_8` – right to delete Domains System log entries.
- Go to the WS_Content directory
- Click on the Domains link in the top navigation
- Click on the Logs link in the navigation beneath that

Any administrator with rights to view or delete log data will be able to see any data that has ever been added, changed or deleted. This is very important. An example is if one administrator adds a new Configuration Variable, say, `MERCHANT_ACC_NUM`. Any administrator with rights to view

log data will be able to see this new Configuration Variable and it’s value. Even if that administrator does not have rights to view the Configuration Variable itself!

Everything that an administrator does is tracked including all additions, edits, and deletions. The complete list of things tracked (this includes all additions, edits or deletions of the thing):

- **Domains System:**
  - Actions:
    - Action Constant
    - Action Description
    - Action Code
  - Configuration Variables
    - Configuration Variable Constant
    - Configuration Variable Description
    - Configuration Variable Value
    - Configuration Variable Server
    - Configuration Variable View Rights
    - Configuration Variable Edit Rights
    - Configuration Variable Type
  - Servers
    - Server Name
    - Server IP Address
    - Server Description
    - Server Type
    - Server WebRoot
    - Server WS_Content
    - Server All Unassigned?
    - Server Login
    - Server Password
    - Server NT Domain
  - Domain
    - Domain Name
    - Domain Description
    - Domain Server On
    - Domain Groups
  - Systems
    - System ID
    - System Constant
    - System Description
  - Colors
    - Color Scheme Name
    - Colors In Scheme
    - Color Name
    - Color Value
  - Text

        - Text Scheme Name
        - Text In Scheme
        - Text Name
        - Text Value
- **Accounts System:**
    - Accounts
        - Account Login
        - Account Password
        - Account Accessible Domains
        - Account Rights Granted
        - Account Rights Revoked
        - Account ADVs
        - Account ADVs Visibility
        - Account Groups
        - Account Ignore Global – Groups
        - Account # Days Active – Groups
        - Account Mailing Lists
        - Account Mailing Opt-Out
    - ADVs
        - ADV Name
        - ADV Description
        - ADV Required?
        - ADV Visible?
        - ADV DataType
        - ADV System
        - ADV Right
        - ADV Group
    - Initial Rights
        - Change to Initial Rights
    - Groups
        - Group Name
        - Group Rights Granted
        - Group Rights Revoked
        - Group Number of Rotations
    - Mailing Lists
        - Mailing List Name
        - Mailing Send – Subject
        - Mailing Send - From Address
        - Mailing Send - From Name
        - Mailing Send – Body
        - Mailing Send – Format

Reading Log Entries

Administrators can choose which Domain to look at log entries from. Only Domains that the administrator has right to view log entries will appear in the list of Domains to manage.

After choosing a Domain, the administrator may choose one of the log types from the log type drop-down list. The log types are shown in the list above.

Administrators may also narrow the search by entering in the `AccountUnq` of a single administrator. Doing so will make the log management page only display actions that administrator performed. This can be helpful when auditing specific users.

Each log entry has several parts. Log entries that "go together" have the same background color. Log entries that "go together" are entries for the same element – entries that affected the same piece of data. For example, sometimes a text constant is edited several times. All of the entries for that specific text constant are grouped together and have the same background color. The next text constant will have a different background color.

Each log entry displays:

- the user who made the change
- the date of the change
- the change type. The change can be one of three things: Update, Delete, or Insertion.
- the logged data. If the type of change was an update, this will store the value of the data as it was before the change. A deletion will store the value as it was before the deletion, and an insert will hold the original value.
- the keys to the data. There can be up to six keys. There can be three text based keys and three numeric keys. Each log type has a different set of keys. The keys are the pieces of data associated with the data to identify and categorize the data. For example, each text constant log entry has three keys:
    - the `TextConst` itself (a textual key). Such as "`ACCOUNTS_ADV_FOOTER1`"
    - the `DomainUnq` (a numeric key) it was changed on
    - the Text Scheme (`SchemeUnq`) (a numeric key) it is in.
- the current data.

The system tries to display additional information with each of the keys. Many of the keys are things like `DomainUnq, SystemUnq`, etc. These id numbers do not mean much to an administrator. So, the system will look the number up and also display the Domain associated with the `DomainUnq`, or the Text Scheme name associated with the `SchemeUnq`.

When the system displays such information, it can often be wrong. The `DomainUnq` or `SchemeUnq` data may be out of date. This is because if a log entry was made prior to a change in Domains, Schemes, etc. the log entry will have the old Domain or Scheme information. So, for example, a text constant may have been changed six months ago. Lets say that back then, the `SchemeUnq` of the Text Scheme that the constant was in was 4 and the name of that scheme was “Website English”. Things have changed and the “Website English” scheme was deleted and a new scheme was created in its place called “Frontpage Chineese”. Now, the log entry will say that the text constant was in Text Scheme number 4 and that Text Scheme number 4 is “Frontpage Chineese”. But, when the log entry was initially created, the text constant was really in “Website English”! **Administrators should understand that the key information may be invalid and to take into consideration the date the action was taken.**

Deleting Log Entries
Administrators may delete log entries. Sometimes the log table (which resides within the SB_Domains database, LogData table) becomes too large and slow. Or, an event happens that creates many log entries.

Administrators should navigate to the log entries on the page to delete log entries the same way they do on the page to view log entries. Log entries can be deleted individually, or the administrator may enter a date to delete log entries. All log entries that occurred before the date will be deleted. This date can also include a time for more granularity.

Custom Log Entries

When developers create new Systems they may need to log actions taken within the new System. Developers can add their own log types by inserting a new row into the LogTypes table in the `SB_Domains` database. The columns in this table include:

- `DomainUnq` – the Domain on which the action took place. *Not necessarily the Domain on which the data resides.* The Domain on which the data resides is a key (see the key information below) – if that data is required.
- `SystemUnq` – the `SystemUnq` of the System the log type is assigned to. Currently only `SystemUnq’s` 1 (Accounts System), 2 (Message Board System – only viewable in the Message Board System log management screen), and 8 (Domains System) are manageable. This is because the log management pages only display from those `SystemUnq’s`.
- `TypeUnq` – this should be a unique ID number for the new log type. Custom log types should begin at 10000 because earlier `TypeUnq’s` are reserved for future use in SiteBeater.
- `Name` – the name of the type that is displayed in the drop-down list on the management screen.
- `Description` – a longer description.
- `CharKey1Name` – the name that is displayed in association with the first character key. See the paragraph below for more information about keys.

- `IntKey2Name` – the name that is displayed in association with the first integer key.
- `CharKey3Name` – the name for the second character key.
- `IntKey4Name` – the name of the second integer key.
- `CharKey5Name` – the name of the third character key.
- `IntKey6Name` – the name of the third integer key.
- `sDB` – the database name where the current data of the log entry can be found. For example, if the log type is for storing changes to account logins, this database would be `SB_Accounts`.
- `sTable` – the table within the `sDB` database that is storing the current value of the log entry. This is used with `sDB` to find the data to display on the log management screen with the log entry. For example, if the log type is for storing changes to account logins, the database would be the `sDB` database, and the table would be Accounts.
- `sColumn` – the column within the table that holds the data for the log type. Continuing the examples from above, the log data for the changes to the login name would be the Login column.

The keys (`CharKey1Name` through `IntKey6Name`) should be chosen carefully. These are the keys used in the query to find the current data for the log entry. Each log entry must have enough information so that, if necessary, the administrator can recreate previous states within the website. For example, lets say we are logging changes to user logins. Each login is dependent on an `AccountUnq`. The login is not dependent on a `DomainUnq` or `SystemUnq`, so we do not need those as keys. But, we do need the `AccountUnq` as a key. Since the `AccountUnq` is an integer, we will use `IntKey2Name` to hold that value. So, in the database, all of the keys will be blank, except `IntKey2Name` will be “AccountUnq”. With this information, the administrator will know which user the change to the login is associated with and can step back in time – per-se – to see all of the states that users login has been in.

After the log type has been added to the database, developers can use the logging API function to log actions. The `DOMAIN_Log()` function is used to save log entries. This function uses the `TypeUnq` for the log type to know which log type to save the data for. See the API documentation for more details on the parameters of this function. An example call to add a log entry for the creation of a new user account would look something like this:

```
Call DOMAIN_Log(36, "", sAccountUnq, "", 0, "", 0, "V", "I", sNewLogin, "")
```

The first parameter is the `TypeUnq`, and the next six parameters are the keys. Notice that only one key has a value – the `AccountUnq` of the new account. The three character keys are blanks and the second and third integer keys are both 0. This function also always assumes that the Domain on which the action is taking place is the current Domain.

Logging too many things can slow the system down. Also, administrators may want to purge the log table on a regular basis, as it gets large.

Servers

The SiteBeater Foundation

18

Using Servers (ALPHA/BETA/LIVE)

Summary:	Using the SiteBeater System with ALPHA, BETA, and LIVE server hardware.
Audience:	Developer, IT
Before You Begin:	Take an inventory of servers and hardware limitations.
Topics Include:	ALPHA/BETA/LIVE Server Configurations Migrating
See Also:	Multiple Servers Configuration Variables

Many companies realize they cannot develop new web pages or systems on live servers. These companies set up alpha, and often beta, servers (primary test and secondary test servers) to develop and test their new systems. SiteBeater can work with such a setup.

ALPHA/BETA/LIVE Server Configurations

Single Domain – Multiple IP Addresses

Each set of servers (ALPHA, BETA, LIVE) has its own IP address (or set of IP addresses – see the section entitled Multiple Servers) and DNS entry. They all use the same Internet domain name. So, ALPHA servers would point to IP address 123.123.123.001, BETA servers would point to 123.123.123.002, and LIVE servers would point to 123.123.123.003. Each of the three would be accessed with http://www.mywebsite.com in the browser.

If the three servers will have different data on them, then there must be at least a MSSQL 2K server for each configuration. In other words, the ALPHA or BETA web server would never look in the LIVE MSSQL database server for their data. They would have their own independent MSSQL servers. SiteBeater needs to have a separate installation for each server for this type of setup.

This is probably not the best use of SiteBeaters capabilities. With this setup, you will not be able to specify which server each Configuration Variable is on since each variable will be always on it's own "LIVE" server (since they have their own installation). You will also not be able to use the SiteBeater migration tool because the tool cannot migrate data from one MSSQL server to another. See the section below on migration for ideas about how to migrate data in this type of setup.

Multiple Domains

Another option is to set up the ALPHA, BETA, and LIVE servers on separate Internet domains. For example, the ALPHA server: www.alpha-mywebsite.com, BETA: www.beta-mywebsite.com, and LIVE: www.mywebsite.com.

To use such a setup, create SiteBeater Domains for each of the three Internet domains. Then, because each is independent, SiteBeater will see each Configuration Variable as having its own possible ALPHA, BETA, and LIVE servers. Therefore, the administrator can use the LIVE Configuration Variables (see the section on Configuration Variables) on each of the three SiteBeater Domains. Or, she can use the ALPHA variables on the alpha machine, the BETA variables on the beta machine, and the LIVE variables on the live machine. But, using the ALPHA Configuration Variables on www.alpha-mywebsite.com would be saying that the alpha server has alpha servers of its own! And, that they use the www.alpha-mywebsite.com Internet domain but different IP addresses.

This technique allows developers to use the SiteBeater migration tool to copy data from one website to another. When the ALPHA website has been tested, it's data can be migrated to the BETA website, and from there to the LIVE website.

Subdomains

It is possible to use one Internet domain name for your ALPHA, BETA, and LIVE systems. Each system has its own subdomain. Create a new SiteBeater Domain for each subdomain. For example: alpha.mywebsite.com, beta.mywebsite.com and www.mywebsite.com. Using SiteBeater to create the IIS entries for the alpha or beta subdomains is not possible because they need independent Host Headers that SiteBeater cannot create. Create the SiteBeater Domains in the SiteBeater administrative pages, and the two new Host Header Names in IIS.

Configuration Variables and migration is treated the same as with Multiple Domains (see above).

Setup	**Pros**	**Cons**
Single Domain – Multiple IP Addresses	• Complete seperation of websites • Added security • Less chance of causing problems on	• Cannot use the SiteBeater migration tool • Required to use separate development

	production websites	machines - Cannot use the same SiteBeater databases (requires independent MSSQL servers) - Must set up separate workstations to access each of the different websites (ie. the workstations DNS must point to either the ALPHA, BETA or LIVE, but only one at a time) - Difficult to setup
Multiple Domains	- Easy to setup - Can use the SiteBeater migration tool - Can use the same SiteBeater databases - Great if you only intend to change data in the database.	- Data is not separate from production websites - Cannot change ASP files in global WS_Content directory without affecting other (possibly live) websites
Subdomains	Same as Multiple Domains	Same as Multiple Domains

SSL
SSL can be used anywhere in the SiteBeater System by purchasing a key and setting it up in IIS. This is usually only done on the LIVE servers.

Migrating

Performing a migration on a server configuration of Single Domain – Multiple IP Addresses, can be a bit scary, but not all that difficult. Migrating from Multiple Domains or Subdomains can be accomplished using the SiteBeater migration tool (see the section on Migration).

Track the files that have changed, been added or removed during development. Copy these files to the destination location – whether it is another server, or just another directory. At the same time (or very quickly), use the MSSQL Enterprise Manager to copy the data from the source server and database to the destination server and database. Enterprise Manager copies all the data in the tables specified from completely different machines. It is necessary to copy the databases of all the Systems changed or added.

These are the databases to migrate according to the data changed:

- Groups – SB_Accounts
- Initial Rights – SB_Accounts
- ADVs – SB_Accounts
- Rights – SB_Accounts
- Users – SB_Accounts
- Mailing Lists – SB_Accounts
- Action – SB_Domains
- Colors – SB_Domains
- Text – SB_Domains
- Domains – SB_Domains
- Servers – SB_Domains
- Configuration Variables – SB_Domains
- Systems – SB_Domains
- Logs (not required) – SB_Domains
- Message Boards – SB_MsgBoard
- Image Galleries – SB_ImageGalleries
- MP3 Catalog – SB_Radio
- News – SB_News

There is no facility built into SiteBeater to migrate a SiteBeater Domain from one MSSQL server to another.

19

Multiple Servers

Summary:	Configure SiteBeater for load balancing technologies and multiple servers.
Audience:	Developers, IT
Before You Begin:	
Topics Include:	Server Setup Server Data Configuration Variables Different Servers/Different Websites Load Balancing Server & IIS Issues
See Also:	Using Servers (ALPHA/BETA/LIVE)

Server Setup

SiteBeater can work with multiple servers in a load-balancing environment. Adding SiteBeater to other servers must be done correctly or data may be lost. Make sure the administrator does not re-run any installation scripts. The installation script for the Foundation deletes all data in the database!

To install SiteBeater on any other server after the first, follow these steps:

1. Make backups of everything.

2. Enter the new server into the list of servers in the SiteBeater database by going to an existing website on an existing server and to the SiteBeater administrative screen Domains->Servers and clicking the "New" button. Enter all of the information for the new server.

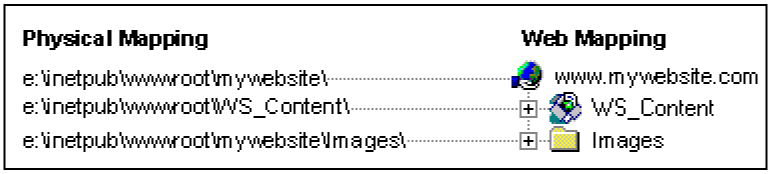
3. Create a website (IIS and DNS entries and directory) on the new server if one is not already set-up (for example: www.mywebsite.com).

4. Copy all of the files from the WS_Content directory of a server with an existing SiteBeater installation to the new server in its corresponding WS_Content directory (usually e:\inetpub\wwwroot\WS_Content\).

5. The ASPMail, ASPImage, and/or TqcRunas DLLs should be installed and registered. Give the DLL files IUSR execute and read access.

6. In the website in IIS, create a virtual directory called WS_Content - at the root of your new websites directory - and point it to the WS_Content directory from step number 4 that holds all the SiteBeater files. Make sure that the read, write, and script properties are checked.

7. You should now have a new website on the new server configured as such: www.mywebsite.com that points to a directory: /inetpub/wwwroot/mywebsite/, that has a virtual directory: /inetpub/wwwroot/mywebsite/WS_Content/, that points to /inetpub/wwwroot/WS_Content/:



Physical Mapping	Web Mapping
e:\inetpub\wwwroot\mywebsite\	www.mywebsite.com
e:\inetpub\wwwroot\WS_Content\	WS_Content
e:\inetpub\wwwroot\mywebsite\Images\	Images

8. Copy the files of the existing website from an existing server to the new server.

9. Go to the website on the new server by setting the load-balancing system to force users to that server. Going to the new server via its IP address will only work if there is a SiteBeater Domain that has that IP Address as its name. Creating a dummy domain with the IP address as its name is a good idea to test the new installation (see the section on SiteBeater Domains for information about creating SiteBeater Domains using IP addresses).

10. The number of the server in the list of SiteBeater servers is in the closing comments of each Action. Check which server is sending each page by looking at the page source (see the section on Actions for more information).

Server Data

Each Server that has a SiteBeater website on it must be entered into the SiteBeater database. Administrators should go to the Domains->Servers page to add new Servers. The initial installation script will enter an ALPHA, BETA, and LIVE Server into the SiteBeater database. This script cannot enter all of the data needed with each Server so an administrator must add the rest before creating any new SiteBeater Domains.

Each Server entry has the Server name, IP address, type, login, password, NT domain, two different paths and a choice of IP addresses to use when creating new domains in IIS.

If a visitor requests a page from an IP address or Server not in the SiteBeater database, an error message will be displayed and SiteBeater will use data from Server number one.

Configuration Variables

Server information is used when creating new SiteBeater Domains. It is also used my SiteBeater to know which Configuration Variable to use on which IP address for which Domain. For example, a company sets up a load-balancing system with three servers and uses the SiteBeater MP3 Catalog. The first server has its MP3s located in a directory that is completely different from server two or three. SiteBeater uses the Server data to know which MP3 path information Configuration Variable to use when a visitor requests a page from any of the three servers. It does this by checking to see which domain they are on, then looking for the IP address of the server they are on. Once it finds the two, it uses that information to find the Configuration Variable reserved for that domain/IP combination.

Another common use is with database connection information. Many times, for security purposes, only a few people hold the database passwords on the LIVE systems. An example of this would be to set the SB_Accounts MSSQL login to use “Accounts_SB” as its password on the ALPHA system. The `SB_ACCOUNTS_PASSWORD` Configuration Variable on the ALPHA server will use this same password. Then on LIVE, the SB_Accounts login would use “jhg9s_hardpw” as the password. The `SB_ACCOUNTS_PASSWORD` Configuration Variable must both be set to “jhg9s_hardpw”. Also make sure to set Rights (`SB_CONF_VIEW_LIVE_DB`) so that only those people who know the LIVE system password can see the Configuration Variables for that SiteBeater Domain. Make sure to also either remove all other users from Groups with those rights, or revoke those rights from Groups they are in.

Administrators can specify which Server each Configuration Variable is on and can make a version of each Configuration Variable for each Server.

Administrators can create copies of each Configuration Variable on as many Servers as they need.

Load Balancing

SiteBeater was created with the ability to run on several web servers concurrently. Many companies have so much web traffic that a single web server cannot bare the load. These companies use load-balancing technologies to split the traffic across several servers. Load balancing configurations often use a different IP address for each machine, but use the same Internet domain.

Currently there is no method of using different subdomains or domains for different web servers of the same website. An example of this type of setup would be to have three web servers in a load-balancing configuration. Each server uses a different subdomain, such as: www1.mywebsite.com, www2.mywebsite.com, and www3.mywebsite.com. Each subdomain brings up the same website and pages are routed to the different servers (i.e. subdomains) via the load-balancing system. This method will not work because SiteBeater uses cookies to track users and their logins.

Instead, use the same domain name on multiple servers (each server with a unique IP address). For example, www.mywebsite.com would be on IP address 123.123.123.001, 123.123.123.002, and 123.123.123.003. Each IP address would be assigned a different machine. The administrator would create three Server entries in SiteBeater – one for each IP address. The Configuration Variables for www.mywebsite.com would then be assigned to each Server as described in the radio example above.

Different Servers/Different Websites

One of the beautiful things about SiteBeater is that each website can be on a different server and still administered from a single point.

Each website can have Configuration Variables assigned to as many Servers as necessary. And, each website can have a directory and IIS entry on many Servers. So, for example, www.mywebsite.com can be on Servers 1, 4, and 9. Then, www.yourwebsite.com can be on Servers 2, 3, 5, and 6. Servers 7 and 8 can be reserved for ten smaller sites each.

The administrator can use SiteBeater to create a new website – www.thissite.com - on Servers 1, 2, and 3. SiteBeater will create the new directories, copy the files, and create the IIS entries – on all three Servers! Then with each Domain and Server combination can be a unique value for each Configuration Variable.

Server & IIS Issues

- If a Server is added, then a Domain added to that Server, then, that Servers IP address or computer name is changed, now, if an administrator tries to delete the Domain, SiteBeater will try and remove it from the wrong machine. SiteBeater will be looking at a new IP address and computer name for the IIS entry and directory. To resolve this, remove Domains first, and then change Server information. Or, do not change Server information at all – unless the machines information is also changed to match. For example, if the IP address of the machine is changed, then update the SiteBeater information for that Server. Never just change the IP address to a different machine – make a new Server entry if a machine is added.

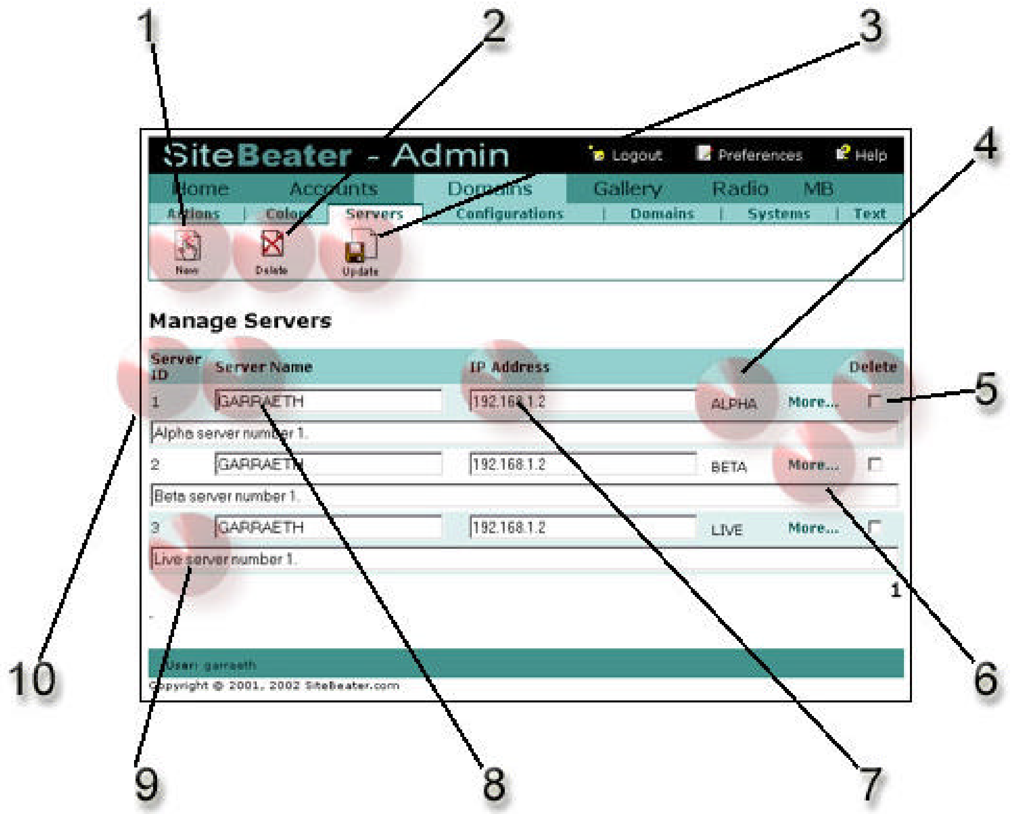
- It is only possible to add, edit or delete a website from IIS or on a different server logged in using NT Authentication.

- The error "-2146893039 (0x80090311)" occurs when the web server has NT Authentication on and it is trying to validate your NT login and password on the web servers NT domain. The server cannot get to the NT domain entered in the NT Authentication login popup. The web server must be on or connected to the correct NT domain. Or, remove it from all NT domains and add it to a Workgroup. Right click on the My Computer icon on the desktop, go to Properties->Network Identification->Properties and make it a member of a Workgroup. Then, in the NT Authentication popup, enter the administrator login and password and leave the domain field blank.

- The administrator will not be able to edit or remove from IIS the Domain that SiteBeater was installed from. This is because SiteBeater cannot know the site name in IIS (it did not create it, so does not know the name). The administrator must make the changes in IIS by hand. SiteBeater can change the name of the Domain in the SiteBeater database, and the administrator can still delete its database entries and directories.

- If an administrator uses SiteBeater to add a Domain in IIS, then deletes it just in IIS, then recreates it again using just IIS, SiteBeater will have lost track of it - with respect to being able to rename or delete it in IIS. SiteBeater will still be able to manage its database data – SiteBeater has just lost track of the IIS site name.

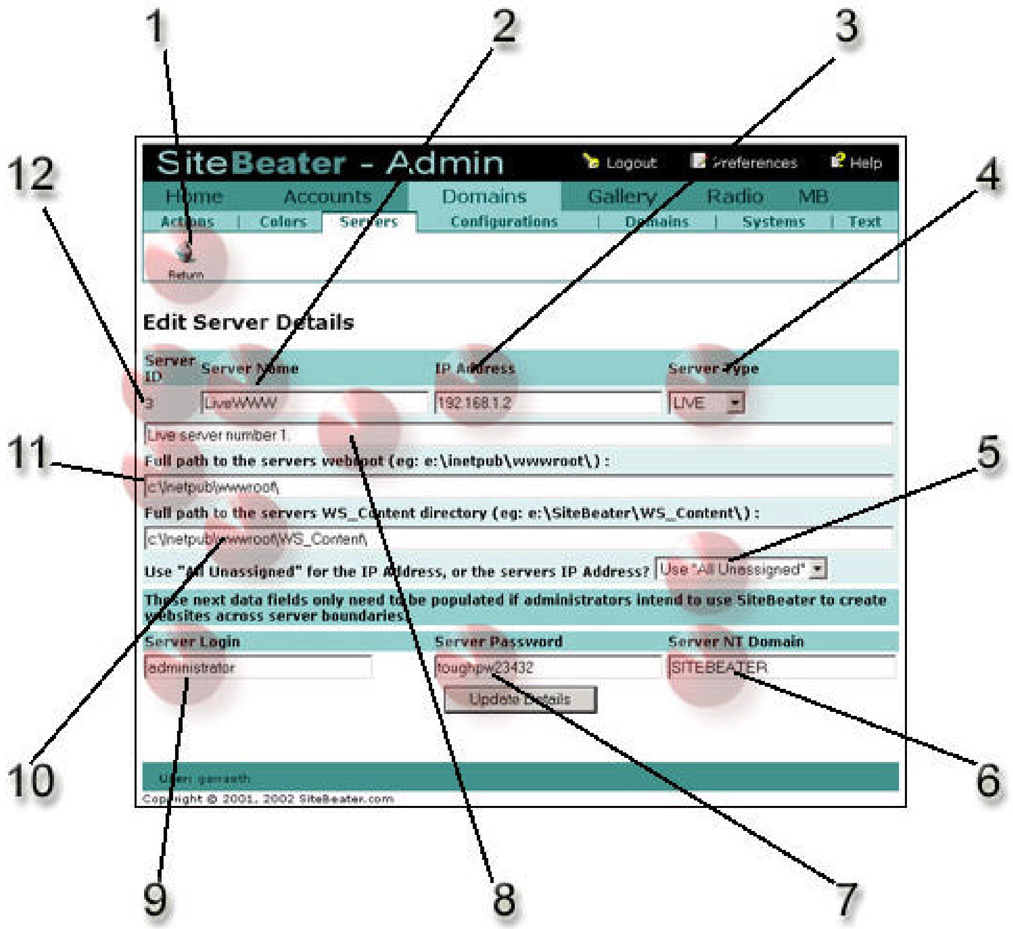
- If the administrator changes the name of a website using SiteBeater in IIS on servers, it will change all the host headers to the new name. For example, if www.mysite.com and mysite.com are host headers, and they are changed in SiteBeater to www.yoursite.com, it will change one of the host headers to www.yoursite.com and the other to yoursite.com.



Manage Servers

Server ID	Server Name	IP Address			Delete
1	GARRAETH	192.168.1.2	ALPHA	More...	☐
Alpha server number 1.					
2	GARRAETH	192.168.1.2	BETA	More...	☐
Beta server number 1.					
3	GARRAETH	192.168.1.2	LIVE	More...	☐
Live server number 1.					

1. Click this button to add a new Server.
2. Click this button to delete Servers with the checkboxes on their line checked (number 5).
3. Click this button to save changes made to Server names (number 8), IP addresses (number 7) or descriptions (number 9).
4. This shows what Type the Server is. Servers can be ALPHA, BETA, or LIVE. This is used to categorize Configuration Variables.
5. Check this box and click the Delete button (number 2) to delete the Server.
6. Click this link to edit the paths, login, password, and NT domain of the Server.
7. This is the IP Address assigned the Server. SiteBeater uses this and the Domain name to determine which Configuration Variables to use when displaying a web page.
8. This is the name of the Server. This the NT computer name. It is used when logging into a server across server boundaries and for being able to identify Servers.
9. This is a description of the Server.
10. This is the Servers unique identifier.



Edit Server Details

Server ID	Server Name	IP Address	Server Type
3	LiveWWW	192.168.1.2	LIVE

Live server number 1.

Full path to the servers webroot (eg: e:\inetpub\wwwroot\) :

c:\inetpub\wwwroot\

Full path to the servers WS_Content directory (eg: e:\SiteBeater\WS_Content\) :

c:\inetpub\wwwroot\WS_Content\

Use "All Unassigned" for the IP Address, or the servers IP Address? Use "All Unassigned"

These next data fields only need to be populated if administrators intend to use SiteBeater to create websites across server boundaries.

Server Login	Server Password	Server NT Domain
administrator	toughpw23432	SITEBEATER

User: garraeth

Copyright © 2001, 2002 SiteBeater.com

1. Click this button to return to the list of Servers.
2. This is the name of the Server. This the NT computer name. It is used when logging into a server across server boundaries and for being able to identify Servers.
3. This is the IP Address assigned the Server. SiteBeater uses this and the Domain name to determine which Configuration Variables to use when displaying a web page.
4. This shows what Type the Server is. Servers can be ALPHA, BETA, or LIVE. This is used to categorize Configuration Variables.
5. Change this to what SiteBeater should put in any IIS entries it creates for SiteBeater Domains. This can be the Servers IP address or “All Unassigned”.
6. This is the NT Domain of the Server. This is used when administering SiteBeater Domains across server boundaries.
7. This is the password to use when administering SiteBeater Domains across server boundaries.
8. This is a description of the Server.

9. This is the login to use when administering SiteBeater Domains across server boundaries. This must be a valid NT account with rights on the other server to either create new directories or write to the IIS metabase.
10. This is the full path to the WS_Content directory. When creating IIS entries, SiteBeater tries to create a Virtual Directory that points to this location – so the new website can access all of the SiteBeater files.

Development

The SiteBeater Foundation

20

Creating WebPages & Custom Systems

Summary:	This explains how to code the ASP pages in a website to use SiteBeater. Also includes, comments on creating custom systems using SiteBeater.
Audience:	Developer
Before You Begin:	Plan and prototype the custom system or website.
Topics Include:	Creating New ASP Files Custom Systems Sample Code
See Also:	

The SiteBeater administrative pages allow a developer to manipulate and control a website created using SiteBeater. This section will explain how to make a website SiteBeater enabled.

A SiteBeater enabled System can be used on multiple domains and has all of the abilities and features of the SiteBeater Foundation.

Creating new ASP files

There is an unlimited number of ways to create SiteBeater enabled ASP pages. Following are two examples. Both require an include statement at the top of each ASP file:

```
<!--#INCLUDE VIRTUAL="/WS_Content/Includes/i_Includes.asp"-->
```

Once this Include statement has been added to an ASP page, *all* variables must be declared – the i_Includes.asp file has a VBScript pragma statement that requires all variables to be declared.

Example 1:

The first way to create the ASP pages is to include all the required functions within the ASP pages. This method is fine if the functions are not being duplicated in each ASP file – or if repeating the functions in each file is not an issue.

Call the subroutine `DISP_Action("My_Top_Nav")`to display any Action data (HTML or ASP). Replace “My_Top_Nav” with the Action name to display. Anywhere before, between, or after the calls to display the Actions, add custom text or code. **At least one Action must be called in a page before opening any other databases or calling any other SiteBeater API functions.** If you have no specific Action you need to call, you can run the `DISP_Action()` subrountine with nothing between the double quotes – like this `Call DISP_Action(“”)`. The DISP_Action() subroutine will open the SB_Domains database for you (which contains all the Configuration Variable information required to connect to all of the other databases) and detect which server the user is on.

The SB_Accounts database will most likely be used to display some sort of user information, or to determine something about users. Call the subroutine `Call DB_OpenAccounts()` to connect to that database. Next, call the `INIT_LoginDetect()` subroutine. This detects if a user is logged in and what rights they have and if they are in any Groups.

Next, to use Color constants from a Color Scheme, or Text constants from a Text Scheme, populate the `iColorScheme` and `iTextScheme` variables by calling the subroutines (the subroutine returns the values of the Configuration Variables passed it – see the API documentation for more information):
`Call DOMAINS_SetColorScheme("ACCOUNTS_COLORS")`
and
`Call DOMAINS_SetTextScheme("ACCOUNTS_TEXT")`.

Change those calls by inserting the color and text Configuration Variables for the Text and Color Schemes used on the new ASP page. The `ACCOUNTS_COLORS` and `ACCOUNTS_TEXT` are used for the SiteBeater Accounts System. The `ACCOUNTS_COLORS` scheme can be used if the color constants in it match the colors used in the new ASP page.

At the end of the ASP file, consider adding a closing Action to display closing navigation. Finally, add the lines to close all the databases that have been opened. In this case: Call DB_CloseAccounts() and `Call DB_CloseDomains()`.

Example 2:
The second method is used when many web pages are opened and closed in the same way. This method requires that the same databases are opened on every page, and that the same color and text schemes are used.

In each ASP file, the first line should be a call to open an opening Action: `Call DISP_Action("My_Top_Nav")`. Next, add whatever HTML, code or other Actions for the page. Finally close off each page by calling the closing Action: `Call DISP_Action(“My_Closing_Nav”)`.

In the My_Top_Nav Action, make several calls to subroutines. First, open the ASP block by using the `cCODE_DELIMITER` value. By default the `cCODE_DELIMITER` is `%PW%`. Within the `%PW%` tags, add the calls to open the required databases (remember that the `DISP_Action()` subroutine automatically opens the SB_Domains database), and the calls to populate the color and text scheme variables:

```
%PW%
Call DB_OpenAccounts()
Call INIT_LoginDetect()
Call DOMAINS_SetColorScheme("ACCOUNTS_COLORS")
Call DOMAINS_SetTextScheme("ACCOUNTS_TEXT")
%PW%
```

(Replace the `ACCOUNTS_COLORS` and `ACCOUNTS_TEXT` with whatever color and text Configuration Variables are needed on the pages.

Add the HTML, code and other Actions to the pages to display the page content. Finally, create a closing Action and add a call to it at the bottom of each page. Within the closing Action, add code to close the databases:

```
%PW%
Call DB_CloseAccounts()
Call DB_CloseDomains()
%PW%
```

Remember to close the SB_Domains database – which you’ve opened indirectly by calling `DISP_Action()` the first time.

Remember to add the include file to each ASP file that will be using SiteBeater functions (Actions, Rights, etc.). From there, open the SB_Accounts database to get account information.

Open page Subroutines and Functions:

```
     Call DB_OpenAccounts()
     Call INIT_LoginDetect()
     Call DOMAINS_SetColorScheme("Configuration Variable")
     Call DOMAINS_SetTextScheme("Configuration Variable")

     Call DISP_Action(“Action Name”)
```

Close page Subroutines and Functions:

```
     Call DB_CloseAccounts()
     Call DB_CloseDomains()
```

Custom Systems

SiteBeater allows developers to create whole systems using the API. Possible systems include: Classified Ad System, eCommerce Systems, Auction Systems, and more.

Generally when creating a new system, developers would first create a prototype of the new system on a test website. New systems should be given their own directory within the WS_Content directory. This allows them to be used on multiple domains.

The system should have a system entry in the Domains->Systems page. This helps categorize Actions, Rights, Configuration Variables, Color and Text Schemes. Remember that SiteBeater System entries are cross-domain and all websites that use SiteBeater will also be able to use the new System.

The system should use color and text constants from Color and Text Schemes the developers create themselves. Each custom scheme should have an accompanying Configuration Variable. When using custom schemes, each ASP page or opening Action should use the code:

```
%PW%
Call DOMAINS_SetColorScheme("Configuration Variable")
Call DOMAINS_SetTextScheme("Configuration Variable")
%PW%
```

Each new system should also have its own Actions. Add these Actions under the new SiteBeater System entry. The Actions should help with navigation on the pages in the new system. Generally there are two sets of navigation: user navigation and administrative navigation. Do not forget to create administrative pages for the new system.

The new system will also require new Rights. Use the API functions to create new Rights, and return user right levels. The rights can be applied to the administrative pages of the new system. If desired, a new Group could also be created for administrators who will work with the new system.

Configuration Variables will also need to be created for the new system. In addition to the Color and Text Scheme variables, each new system will need several other variables. If the new system connects to a database, login, and password can all be held within Configuration Variables. You might also want to look at what Configuration Variables each of the different SiteBeater systems have. Maybe email variables or display formatting variables.

New systems may also have ADVs associated with them. It is generally a good idea to keep all user information in ADVs – even if the new system has its own database. Using ADVs gives the developers the advantage of being able to use the

SiteBeater API functions, share data across domains, and keep the data in one place for backup, replication, and such. On the other hand, developers may want to keep some user data separate. For example, very confidential information could be kept on a separate, secure database server reserved for such data.

The Actions, Color and Text Schemes, Rights, and Configuration Variables that are created on one SiteBeater Domain will not appear on any other existing Domain. New SiteBeater Domains can use the Domain the system was created on as their Source Domains to gain a copy of the data for itself. Developers will have to create installation scripts for the other Domains – to add the database data to those Domains. It is possible to use Microsoft's Query Analyzer to copy the data from one Domain to another. Or, enter the data on each Domain that will be using the new system. The SiteBeater migration tool can be used to copy Actions, Color and Text Schemes, and Configuration Variables to the other domains. Using the migration tool is not recommended if there are many domains because it will take a long time do copy all of the data by hand – create a small installation script to migrate the data instead.

Programming Notes:

- The `sReturnPage` variable can be used with the login page. When linking to the login page, if the `sReturnPage` variable is in the QueryString or in a form, the user will be sent to the URL the variable is set to instead of the referring document. This is great if, after a user has logged in, the need to be sent to a specific web page.

- ASP variables, subs and functions may be used within Actions. These have the same scope as they would if the `DISP_Action()` call that "run" the Actions were replaced with the Actions code itself.

- Actions can also retrieve data from the QueryString, forms, cookies, application variable, and session variables.

- The `DOMAIN_Conf(sConfConst)` function returns the value of the Configuration Variable passed it. See the API documentation for more information and examples.

- The `ACCNT_ReturnRights(sRightsLvl)` function returns `TRUE` if the user currently logged in has the right level passed it. Again, see the API documentation for more information and other functions.

- Groups do not usually need to be explicitly programmed for. When code checks for user rights, it automatically checks for Group rights that user may also have.

- Make sure to program with multiple domain functionality in mind. When specifying path information for files, use Configuration Variables instead of hardcoding the path, as the path may be different from domain to domain or server to server. Also if different people administer Domains, those people might like to do some things differently. Allow the administrator some leeway by making some things Configuration Variables instead of hard coded.

- In addition to creating new Actions, these Actions in the Accounts System should also be changed: `SB_ACCOUNTS_LOGIN_OPEN`, `SB_ACCOUNTS_LOGIN_CLOSE`, `SB_ACCOUNTS_NEWACCOUNT_OPEN`, `SB_ACCOUNTS_NEWACCOUNT_CLOSE`, `SB_ACCOUNTS_USERDATA_OPEN`, `SB_ACCOUNTS_USERDATA_CLOSE`. These Actions are used on the login and user data pages that site visitors see.

- Use the SiteBeater API function to send emails.

- System entries in SiteBeater are cross-domain. A new System created on the Domains->Systems page will appear on all Domains for all administrators with rights to view Systems.

- The `DOMAIN_Color(sColorConst)` function returns the value of a color constant in the current Color Scheme. The current Color Scheme is stored as an integer in the `iColorScheme` variable. See the API for more information on the function.

- The `DOMAIN_Text(sTextConst)` function returns the text in a text constant in the current Text Scheme. Just like colors, the current Text Scheme is stored as an integer in the `iTextScheme` variable. The API has more information.

- There is an i_Custom.asp file in the /WS_Content/Includes/ directory that is included in the i_Globals.asp file, which, in turn is included in <u>all</u>

SiteBeater Systems. Add custom subroutines and functions to this include file and they will be accessible by all SiteBeater Systems. The i_Custom.asp file will never be overwritten by updates.

- Users can set ADV information as "Visible" or "Hidden". When developing custom systems, or using ADVs in Actions, the `ACCNT_ReturnADV()` API function will only return the ADV data if the current user is the owner of the ADV, or if the ADV has been set as Visible by the owner of the ADV.

  Sometimes you need to read the ADV data even if it is hidden to the general public. One example of this is when a user sends an email to the administrator in the Message Board using the "alert administrator of a offensive post" feature. What happens if the administrator set their email address as a hidden ADV? Even though the administrator is not the user logged in, we still need the email address to send the mail. Use the "OVERRIDE" keyword in the `ACCNT_ReturnADV()` function in place of the sViewLvl parameter to have it return the value of the ADV even though that ADV is supposed to be Hidden.

Sample Code

- The Action in the Domains System called `SB_LOGIN_LINK` can be used to display a link to either log in or log out. The link changes depending on if the user is already logged in or out.

- The Action in the Domains System called `SB_PREFERENCES_LINK` can be used to display a link to edit user preferences or create a new account. The link changes depending on if the user is logged in or out.

- The Action in the Domains System called `SB_ COLORSCHEME_CHANGE` can be used to change the current Color Scheme to the Scheme number on the QueryString by the name of iColorScheme. It will also read it from a form or cookie. This changes the iColorScheme global variable and only changes it if something is passed it.

- The Action in the Domains System called `SB_TEXTSCHEME_CHANGE` can be used the same way as the `SB_COLORSCHEME_CHANGE` Action, but for Text Schemes. It looks for iTextScheme on the QueryString, from a form, or cookie. It sets the iTextScheme global variable only if something is there.

- The Action in the Domains System called SB_PRODUCT_DISPLAY gives a simple example of how a product display page could be created.

21

Global Variables

Summary:	This is a list of useful variables used by the SiteBeater Systems. Use these when creating custom Systems.
Audience:	Developer
Before You Begin:	Understand how to code custom Systems using the SiteBeater API.
Topics Include:	
See Also:	Customizing Existing Systems, Creating Custom Systems

Global variables are used in the SiteBeater Systems and are available for use in custom Systems and code.

- `SITEBEATER_VERSION` – the version of SiteBeater Foundation. This is displayed in the web page source with the Action header comment (see section on Actions).

The `MSWC.BrowserType` object fills these next few browser detection variables. Reference Microsoft's documentation for more information on what each variable contains.

- `Browser`
- `Version`
- `Tables`
- `Frames`
- `Cookies`
- `BgSounds`
- `JavaScript`
- `VbScript`
- `ActiveX`
- `Channels`

Display Globals – these variables were added to help with the creation of custom Systems. For example, with the `sPageTitle`, it is possible to type text in for the title of each page, or use a text constant to print it out. On the other hand, if the same Action opens several pages that have different titles, the `sPageTitle` can set the title to something different on each page.

- `sPageTitle` – optionally use this in the Actions that opens each web page to set the title of the page. Print it between the `<title></title>` tags. Set it in the ASP before calling `DISP_Action()`.

- `sStyle` – optionally use this in the Actions that opens each web page to append additional Stylesheet codes. Print it between the `<style></style>` tags. Set it in the ASP before calling the `DISP_Action()`.

- `sHeader` – optionally use this in the Actions that opens each web page to append additional head codes. Print it between the `<head></head>` tags. Set it in the ASP before calling the `DISP_Action()`.

- `iDomainName` – this is set in the `i_Domains.asp` file in the `DISP_Action()` subroutine. Use this any time the `DomainUnq` (from the `SB_Domains` database, `DomainInfo` table) of the current SiteBeater Domain is needed.

Account Globals

- `iLoginAccountUnq` – this is the `AccountUnq` (`SB_Accounts` database, `Accounts` table) of the user who is logged in. This variable is set in the `i_Accounts.asp` file, in the `INIT_LoginDetect()` subroutine. If the surfer is not a member, this value is `NONMEMBER_ID` (-1).

- `bHasAccount` – this is either `TRUE` if the user is a member, or `FALSE` if they aren't logged in. This variable is set in the `i_Accounts.asp` file, in the `INIT_LoginDetect()` subroutine.

- `sGroupType` – this is the Group the logged in user is in. If the use is not logged in, or not in any Group, this variable is blank. Otherwise, it's an "S" if the user is in at least one Standard Group, or an "R" if they are in a Rotational Group. Users cannot be both in a Standard and Rotational Group at the same time.

These variables help quicken the processing of the `ACCNT_ReturnRights()` function as it is called many times.

- `sGlobalRotations` – If the user is in a Rotational Group, this is the number of rotations.

- `sGlobalDateBegin` – If the user is in a Rotational Group, this is the date the rotations began or begin.

- `sGlobalGroupUnq` – If the user is in a Rotational Group, this is the `GroupUnq`. We can't get this for Standard Groups because there might be many of them.

- `sGlobalPosition` – If the user is in a Rotational Group, this is the position of the user in the rotation.

- `IGlobalNumDays` – If the user is in a Rotational Group, this is the number of days the current login works.

These are generic global constants that are used throughout the Systems, and for use in custom Systems.

- `VOID` = -1
- `EQUAL` = 1
- `GREATER_OR_EQUAL` = 2
- `RIGHTS_NORIGHTS` = "-1"
- `NONMEMBER_ID` = -1

These three constants are used when checking which server we are on in `i_Domains.asp`, in the `DOMAIN_CheckIP()` subroutine.

- `LIVE` = 1
- `BETA` = 2
- `ALPHA` = 3

These two constants are used when determining if an account can access all SiteBeater Domains, or only it’s Home Domain.

- `HOMEONLY` = -1
- `ALLDOMAINS` = -2

- `SERVER_VERSION` – this can be 1, 2, or 3 for LIVE, BETA, ALPHA. This is actually the server *type*. It can be used to detect which type of server the user is currently on. For example, say you would like to send an email to an administrator from a certain webpage. On your ALPHA and BETA servers, you can use this variable to have that webpage display the email on screen and not actually send it. Then, on the LIVE server, the webpage will go ahead and send the email.

- `iTextScheme` – this should be set to the Text Scheme ID number to use when displaying text constants. This variable is used in the `DOMAIN_Text()` function in `i_Domains.asp`. The scheme numbers can be found in the `SchemeUnq` column in the `SB_Domains` database, `TextSchemes` table. Make sure to check that the `DomainUnq` (SiteBeater Domain ID number) is for the correct domain. The scheme number can also be found on the Text Scheme management page, next to the scheme name in the scheme drop-down list.

- `iColorScheme` – this is the same as `iTextScheme`, except for colors (`ColorSchemes` table, etc).

- `sServerCV` – this is set to the `ServerUnq` of the server that is sending the current page. This is used in each Actions closing comment. It is only displayed on BETA and LIVE servers.

These two arrays are used in the function (`DOMAIN_Link()` in `i_Domains.asp`) to add QueryStrings to links and hidden fields to forms. A maximum of twenty variables and their values can be used in this function.

- `aVariables(20)` – this is the list of variable names

- `aValues(20)` – these are the values of the variables in the `aVariables` array.

Miscellaneous variables:

- `GLOBAL_DEBUG` – Set this to true to turn on all debugging output from all ASP pages, or set it individually in each page. This may be set to any value in the QueryString to turn it on, and will then be stored in a Session variable. Kill the Session variable by putting this on the QueryString again with no value.

- `sLOGGING_ENABLED` – this holds the status of the `LOGGING_ENABLED` Configuration Variable. Its used to speed up the process of logging – so the database doesn't have to be checked multiple times from the same ASP page.

- `dColors` – This is the ASP Dictionary Object that holds cached color values from the current Color Scheme.

- `dSpecificDomain` – This is similar to `dColors` in that it caches access rights when the `ACCNT_SpecificDomain()` function is called.

- `dSystemRights` – As with `dSpecificDomain`, this also caches access rights, but when the `ACCNT_SystemRights()` function is called.

22

Upgrading & Patching

Summary:	Upgrading and patching each SiteBeater System.
Audience:	Administrator, Developer
Before You Begin:	Backup all code and databases.
Topics Include:	Special Upgrades
See Also:	Foundation Setup

Upgrade and patches for the SiteBeater Foundation and other Systems will be released periodically. When upgrading and patching, the SiteBeater websites may be down (or parts of them) depending on what has been changed. Prepare ahead of time for each upgrade. Be able to quickly revert to the original version in case of unforeseen problems. If the facilities are available, perform the upgrade on an ALPHA and then BETA computer first. Test websites on those computers to make sure everything is working. Once everything is working, perform the update on LIVE machines.

Before beginning the upgrade or patching process, backup all relevant files and databases. If developers have changed any of the SiteBeater files, upgrading may or may not work – depending on what was changed in the upgrade. Generally, upgrading involves overwriting all the existing files with the new ones in the upgrade. If developers have changed any of the files that have been changed by the SiteBeater team, upgrade changes must be integrated by hand.

Each upgrade and patch comes in the Upgrades directory of the Foundation or System being upgraded. There are two parts to most upgrades. The first part is the actual files that have been changed. Those files will be located in their respective directories in a replica of the System's directory structure. For example, a new version of the `/WS_Content/Includes/i_Globals.asp` would be in the `/WS_Content/Includes/` directory within the upgrade directory. If developers changed any of these files, they will need to look at the new version and integrate the changes into the custom version of the files.

Each upgrade script is only for one version. Upgrading from version 1 to version 4, requires that the developer copy all the files from each of the three upgrades over the originals. It is not possible to simply copy the files from version 4 because each version has a different set of files. All of the files from version 2, 3, and 4 must be copied or integrated.

The second part of each upgrade is running the database update script. As with upgrading files, each script will only make the changes for a single upgrade. In other words, there is no upgrade script that will upgrade the databases from version 1 to version 4. All upgrade scripts will only upgrade from version 1 to 2, then from 2 to 3, etc. To upgrade from version 1 to version 4, requires running all three upgrade scripts.

The upgrade scripts are in the `_Upgrade` directory of each System. After an upgrade script has been run, it should be deleted so there is no chance of it being run a second time.

The current version of SiteBeater can be found printed in each pages source in each Actions opening comment.

Special Upgrades

i_OpenDomain.asp

If the `i_OpenDomain.asp` file is ever upgraded, first copy the database connection string from your original `i_OpenDomain.asp` file into the new `i_OpenDomain.asp` file. Look in the file for the line under this comment:

```
'***********SB_DOMAINS CONNECTION STRING BELOW******
```

This is the only file that requires a manual edit.

23

Configuring Edit-On Pro

Summary:	Edit-On Pro WYSIWYG editor can be used to edit text in all SiteBeater Systems.
Audience:	Administrator
Before You Begin:	Purchase and install Edit-On Pro
Topics Include:	
See Also:	

Administrators should only read this section if they have purchased the Edit-On Pro Java WYSIWYG HTML editor (http://www.realobjects.de).

Copy the files into the `/WS_Content/Edit-On/` directory preserving directory structure and **not** overwriting any existing files (SiteBeater has customized versions of some of the Edit-On Pro files). Visit the RealObjects.de website to purchase or request a trial version.

Configuration Variables

- `USE_EDIT-ON` – Set this to “YES” to use the editor when editing long text data in Text Schemes, and when editing Actions. Set it otherwise to use the standard HTML text box. Remember, this must be set to “YES” to use the editor in any other System (Message Board, Radio, Image Gallery).

- `EDIT-ON_LICENSEKEY` – The license key for this domain. Populate this only if not using the standard licensekey.xml file. Leave blank to use the file.

- `EDIT-ON_SPELLCHECKPROPERTIES` – Set to the spell check properties file to use. Change according to the language. See the Edit-On Pro documentation.

- `EDIT-ON_IMAGEROOT` – Imageroot value from Edit-On Pro. See the Edit-On Pro documentation. This can be set to a single image server if desired (multiple web servers/one image server type of setup).

- `EDIT-ON_TEXT` – The Text Scheme Unq used with Edit-On. Must be a default SiteBeater Text Scheme or scheme created using a default SiteBeater Text Scheme.

- `EDIT-ON_IMAGEPROXYURL` – Use with the `EDIT-ON_IMAGEROOT`. The IP address/Domain name (without the "http://") of the server hosting all images (e.g. 192.168.1.1) – or leave blank to use images from different domains.

Each System (except the Accounts and Domains Systems) has its own Configuration Variables that can be set differently for that System. To use Edit-On in a System, make sure the `USE_EDIT-ON` Configuration Variable is set to "YES" and the System's `USE_EDIT-ON` Configuration Variable is set to "YES" as well.

When Edit-On is being used in the Domains System, the long text of text values within Text Schemes can be edited in the editor. Actions can also be edited with Edit-On Pro. In the Radio System, play lists can be edited using the editor. In the Message Board, creating new threads, editing threads, replying to threads and editing signatures can all be changed using Edit-On Pro. Finally, in the Image Gallery, editing gallery descriptions can be done using Edit-On Pro.

Edit-On Pro Configuration Variables

- `USE_EDIT-ON`
- `EDIT-ON_LICENSEKEY`
- `EDIT-ON_SPELLCHECKPROPERTIES`
- `EDIT-ON_IMAGEROOT`
- `EDIT-ON_TEXT`
- `EDIT-ON_IMAGEPROXYURL`
- `MSGBOARD_EDIT-ON –` Set to "YES" to use in the Message Board
- `MSGBOARD_EDIT-ON_SPELLCHECKPROPERTIES`
- `MSGBOARD_EDIT-ON_IMAGEROOT`
- `MSGBOARD_EDIT-ON_TEXT`
- `MSGBOARD_EDIT-ON_IMAGEPROXYURL`
- `MSGBOARD_EDIT-ON_WIDTH`
- `RADIO_EDIT-ON –` Set to "YES" to use in the Radio
- `RADIO_EDIT-ON_SPELLCHECKPROPERTIES`
- `RADIO_EDIT-ON_IMAGEROOT`
- `RADIO_EDIT-ON_TEXT`
- `RADIO_EDIT-ON_IMAGEPROXYURL`

24

Configuring ASPMail, ASPQMail & CDONTS

Summary:	General information about the email systems that can be used with SiteBeater.
Audience:	Administrator, IT
Before You Begin:	Consider the options between the different email systems.
Topics Include:	
See Also:	

SiteBeater can use ASPMail, ASPQMAIL or CDONTS to send emails. The `EMAIL_REMOTEHOST` Configuration Variable must have a value on the Server for the email functions to work. Generally, the email functions do not even appear on the web pages if the `EMAIL_REMOTEHOST` variable is blank.

Each Server that has SiteBeater installed on it must have one of the three email systems installed to send emails. If none are installed, leave the `EMAIL_REMOTEHOST` variable blank.

Please make sure that the email remote host is your, or a public remote host.

If there are a lot of emails being sent, CDONTS may be the email system to use. With CDONTS, it is possible to set up a separate email machine on which the emails will be queued and sent. The ASPMail systems use the web server to send the emails.

Email Configuration Variables

- EMAIL_REMOTEHOST – set this to the remote host (mail server) used to send emails. Leave this blank to disable emails – email form fields and links will be removed from SiteBeater Systems.

- EMAIL_TYPE – set this to either “ASPMAIL” or “CDONTS” depending on which email system to use. To use either ASPMAIL or ASPQMail, set this to “ASPMAIL”. This is set to “ASPMAIL” upon installation of the SiteBeater Foundation.

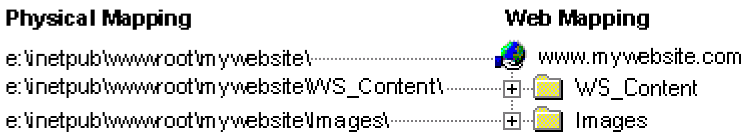
Appendices

The SiteBeater Foundation

A

Troubleshooting

Summary:	Help with common problems.
Audience:	Developer, Administrator, IT
Before You Begin:	
Topics Include:	
See Also:	

Question or Problem	Answer or Solution
The installation script gives me an error about permissions when I try and create a new website.	If the permissions are reset after the initial installation of the Foundation, SiteBeater cannot programmatically create new domain directories (from the Domains->New Domain page) unless TqcRunas is installed. You can either install TqcRunas or create your new domain and not copy files from an existing domain. This will create all of the database entries required for the new domain and you will have to copy the files by hand.
I am hosting my website and cannot create a Virtual Directory as directed in the installation instructions. Can I still use SiteBeater?	Yes. Instead of copying all of the SiteBeater files to a WS_Content directory outside of your websites directory, copy them into a WS_Content directory within your website directory. This image illustrates the layout:  Physical Mapping Web Mapping e:\inetpub\wwwroot\mywebsite\ www.mywebsite.com e:\inetpub\wwwroot\mywebsite\WS_Content\ WS_Content e:\inetpub\wwwroot\mywebsite\Images\ Images This method requires that for each website you want to use SiteBeater with; you have to copy all of the WS_Content

	files into the websites directory. If you make a change to the SiteBeater source files (or apply an update), you have to apply the changes to all copies of the SiteBeater files. This also means that you can make change to the SiteBeater files to completely customize each website you have. So, it does have some beneficial side effects.
I am reinstalling SiteBeater or one of its Systems and it keeps telling me that it cannot complete the installation because one of the databases is in use. What do I do?	Each time the installation scripts run for the SiteBeater Foundation or any of the additional Systems, it checks to see if the databases and logins already exist. If they do, it deletes them. It does this to make sure that the database server is completely wiped of all previous (and possibly corrupt or incorrect) information is gone. The scripts will automatically create the logins and database as needed (this only applies to installations where the database server is on the same machine as the web server). To avoid the error you are getting, make sure there are no connections to the databases that go with the System you are trying to install. For the Foundation, that would be the SB_Domains and SB_Accounts databases. Connections can be opened by web browsers, Enterprise Manager, Query Analyzer, or other third party applications that connect to the database (such as erWin® or Crystal Reports®). Make sure these applications are not connected to the databases. These applications (including browsers) can be connected across networks as well, so if you have closed all connections locally, you might want to check computers on your network. If all of those computers are disconnected, the problem may be that outside users are hitting web pages that use the databases – fixing this means making sure no users can hit pages accessing the databases. Do this however you see best. After you have disconnected everything, you may have to wait a few minutes for the database server to reset itself. If worse comes to worse, you can try to manually remove the databases in Enterprise Manager.
I am getting this error on my new website: "*Critical Error: Domain name not found in database. Quitting.*"	This error generally appears when a new SiteBeater Domain has not been created for the Internet domain. This error could also occur when accessing a website using its IP address (for example, http://123.123.123.123 instead of http://www.mysite.com). If, when creating the SiteBeater Domain, the administrator used "www.mysite.com" as the Domain name, the users must

	type that exact text (case insensitive) in the browsers address bar. To use the IP address, change the SiteBeater Domains name to the IP address numbers (“123.123.123.003”).

B

Security Issues

Summary:	Server and database security in relation to SiteBeater Systems.
Audience:	IT, Administrator
Before You Begin:	The server must be setup with IIS, MSSQL2k, and a DNS entry for the website.
Topics Include:	Installation Issues Separating Server Access NT Authentication Multiple Servers IIS Administration
See Also:	Rights, Initial Rights, Groups, and Configuration Variables, SiteBeater Domains, Multiple Servers

Security is always important. The security issues addressed in this section are related to the web and database servers. SiteBeater System security issues are addressed in the Rights, Initial Rights and Groups sections.

Installation Issues

When first installing the SiteBeater Foundation, specific rights will need to be granted to the `IUSR_computername` account in several directories. However, after installing the Foundation, the majority of these rights may be revoked. This list shows the directories, their rights, and whether or not those rights need to be revoked:

- `/WS_Content/Includes/` - modify – yes, revoke these rights
- `/WS_Content/` - read, execute – no, leave these rights
- Installation directory (e.g. e:\inetput\wwwroot\) – modify – only leave these rights if the SiteBeater System is needed to create directories and copy files while creating new SiteBeater Domains. This right may be revoked if TqcRunas is used.

- The website’s first directory – modify – yes, revoke these rights

After installation is complete (of any SiteBeater System), the installation directory should be removed or renamed. If it is not, anyone who knows the information on the form can reinstall the System and wipe out any data.

Separating Server Access

In the Multiple Servers section, an example is shown of a set of database passwords for a three server configuration: ALPHA, BETA, and LIVE. Reading this should help limit LIVE environment exposure to corruption.

In addition to the database passwords, a separate cCODE_DELIMITER (see the section on Configuration Variables and Actions) may be wanted for LIVE servers. The cCODE_DELIMITER is amazingly powerful. Any party who knows the value of this variable can run *any* ASP command or program on a web server, this code enables a developer to connect to any database if they know the database password. Files can be erased, uploaded and more. On the other hand, cCODE_DELIMITER changes are very difficult to manage. Developers must change the Configuration Variable value and all occurrences of it in Actions they have coded.

A System and Action may be developed and then before migrating them (see the section on Migration), the individual who knows the cCODE_DELIMITER on the LIVE servers may evaluate them and replace the ALPHA delimiter with the LIVE one.

Administrators should manage who has rights to use ASP in Actions by setting `SB_MACT_USE_ASP` user rights.

NT Authentication

The administrative areas of the SiteBeater Foundation are in two directories. For additional protection, set up NT Authentication on the `/WS_Content/Domains/` and/or `/WS_Content/Accounts/` directories.

NT Authentication is also required to create IIS entries on any server. It is also required if administrators wish to use the TqcRunas DLL to add websites to servers other than the machine serving the administrative web pages.

Multiple Servers

SiteBeater can administer domains across server boundaries. Administers will want to review the sections on SiteBeater Domains and Multiple Servers.

It is wise to reserve a single internal domain or IP address for administering SiteBeater websites and domains across server boundaries. This domain must be protected with NT Authentication. In IIS, set that domain's “Application Protection” to “High (Isolated)”. Then remove “Anonymous access”. The server then may be added to an NT Domain with the other web servers. Administrators must then log into the domain using an NT administrative account.

Each directory on each server that will have domain directories and files added to it must also have write access for the `IUSR_machinename` unless the TqcRunas DLL is used. If this is the case, an account with write access to those directories and files should be used to make changes.

The TqcRunas DLL (http://www.quimeras.com) must be registered on the server that the administrative domain is on. This DLL only needs to be registered on the one machine. This TqcRunas DLL poses a huge security threat because anyone who gains access to it can log onto any server using any account. The administrator may also want to restrict IP address access to all of the domains on this machine (because all domains on the machine can access the DLL).

IIS Administration

SiteBeater can add, edit and remove domains from IIS as they are added, changed and removed from the SiteBeater System.

As explained in the Multiple Servers section above, for administrators to change IIS entries, the domain they are doing it from must have NT Authentication. In IIS, set that domain's “Application Protection” to “High (Isolated)”. Then remove “Anonymous access”. Administrators must then log into the domain using an NT administrative account.

As long as IIS Administration is being done on a domain on the machine changes are being made to, the TqcRunas DLL does not need to be registered (they still need NT Authentication). TqcRunas is only needed to make changes across server boundaries.

Please be aware that administering IIS or administering across server boundaries can be a great security risk. Such administration is normally done behind strong firewalls.

C

Database Issues

Summary:	Database setup and information for the Foundation.
Audience:	Administrator, Developer
Before You Begin:	Install MSSQL Server
Topics Include:	
See Also:	Foundation Setup

SiteBeater uses the MSSQL database to store website and user information. The information within this section should help set up and maintain the databases.

The MSSQL server on which SiteBeater databases are to be installed must have "SQL Server and Windows Authentication". To set this:

1. Right click on the server in Enterprise Manager.
2. Click Properties.
3. Click the Security tab.
4. Click the radio button for "SQL Server and Windows".

Two databases must be created for the Foundation:

- c. SB_Accounts
- d. SB_Domains
- e. The database logins are (these names are the default – you may change them when installing the Foundation):
    - i. SB_Accounts – make this the owner of the SB_Accounts database, and make the SB_Accounts the default database. This password should be in the `SB_ACCOUNTS_PASSWORD` Configuration Variable.
    - ii. SB_Domains – make this the owner of the SB_Domains database, and make the SB_Domains the default database. This password should be in the /WS_Content/Includes/i_OpenDomain.asp file. It is the last parameter used with the `DomainsConnection.Open` statement.

API functions can be used to open and close database connections:

- `DB_OpenDomains()` – sets both the `DomainsConnection` and `DomainsCommand` variables. These variables are also set when the `DISP_Action()` subroutine is called. Generally the `SB_Domains` database is opened and connected to through the `DISP_Action()` subroutine.
- `DB_OpenAccounts()` – sets the `AccountsConnection` and `AccountsCommand` variables.
- `DB_CloseDomains()`
- `DB_CloseAccounts()`

If the `SB_Domains` login and/or password is changed in the database, it must also be changed in the `/WS_Content/Includes/i_OpenDomain.asp` file in the `DB_OpenDomains()` function on the `DomainsConnection.Open` line. The `SB_Domains` database password must be changed in the actual file because that is the first place the database is accessed.

If the DB_Accounts database login and/or password is changed, then the corresponding Configuration Variables (for each server) must also be changed:

- SB_ACCOUNTS_SERVER
- SB_ACCOUNTS_LOGIN
- SB_ACCOUNTS_PASSWORD

D

Installing On Hosting Services

Summary:	Information for those wishing to install and use SiteBeater on a third party hosting service.
Audience:	IT, Developers
Before You Begin:	Contact your hosting service and let them know you would like to install SiteBeater.
Topics Include:	
See Also:	Foundation Setup

If you are using a third party to host your website, please read this.

SiteBeater is very powerful and can be complicated. Your must understand (or have someone available) how to use a Microsoft SQL server and programming in ASP before downloading or purchasing SiteBeater. Unlike other companies who use the ignorance of their customers as a marketing tactic, we want to make sure you understand what you are buying.

Hosting Requirements

Databases

To enhance the scalability and power of SiteBeater, we chose to use MSSQL. In addition to using high-end database software, we also chose to separate the system into **five** databases! This way, when your databases grow too big for one machine, you can separate each out onto individual machines. Or, if you need more speed, you can put each database onto its own machine. This also helps with design and modularity issues.

If you decide to either download or purchase SiteBeater and use it with a hosting service, you must be sure your hosting service will allow you to use it. Your hosting company must allow you to set up at least TWO MSSQL databases and

TWO database logins. Both are used for the Foundation. One is for the Domains System and the other for the Accounts System. <u>These are required.</u>

If you decide to install the Image Gallery System, you will need another database and login. If you want to install the Message Board System, you'll need yet another. And so on. If you install all additional Systems, you will need to be able to install–on your hosting service–a total of six databases and SQL logins.

ASP Files

The SiteBeater files are usually placed in a directory accessible to all websites on a web server. This location is most often in a directory above your websites directory. Some explanation might be required here: Generally hosting companies allow you to FTP into one directory and it's subdirectories. That one directory is <u>your</u> website's directory. They do not let you go "up" into the directory "above" your website's directory. If they did, you would be able to see all of the other websites directories, and possibly go into them and change the data there! With SiteBeater, the WS_Content (all of the SiteBeater files are within the WS_Content directory) directory is generally placed into the directory that holds all of the website directories! If your hosting company only allows you to go to your websites directory, you won't be able to create a WS_Content directory in its usual location.

If you can place the WS_Content directory in the directory with all of the other website's directories, you can then create a "virtual" link to it in your website's IIS entry called a "Virtual Directory". Some hosting companies might allow you to create and have your own Virtual Directory. Many do not. If your hosting company will not create a Virtual Directory for you, you cannot put the WS_Content directory into the same directory as all of the other website directories, nor can you create a Virtual Directory pointing to it. Instead, copy the WS_Content directory into *your* websites directory–at the root. Your website won't know the difference. But, you will only be able to use SiteBeater with your one website.

If you can both: 1. Copy the WS_Content directory into the directory with all of the other websites directories, and 2. Create a Virtual Directory pointing to the WS_Content directory, GREAT! Install SiteBeater that way.

If you cannot copy the WS_Content directory into the directory with all of the other websites directories, or if you cannot create a new Virtual Directory pointing to it, then you will have to use an alternate installation method. **Instead**, copy the WS_Content directory into your websites directory–at the root. Your website won't know the difference. But, you will only be able to use SiteBeater with your one website. If you want to use SiteBeater with many websites without a Virtual Directory, you can copy the WS_Content directory into each individual website directory. Each website will use the same SiteBeater databases, but they will use their own copy of the SiteBeater files.

Active-X

If you install the MP3 Catalog System, you must install the ID3 tag reading Active-X object. Some hosting companies might allow you to install a foreign component on their system. There is no way around this.

If you want to use ASPImage (optional) with the Image Gallery, your hosting service must either install it for you, allow you to install it (not likely), or they must already have it installed (many do).

It is very doubtful that any hosting service will allow you to install TqcRunas (also optional). TqcRunas allows you to manipulate data and IIS entries on multiple servers. This is a huge – and usually unacceptable – security risk for a hosting service.

Emails in SiteBeater require CDONTS, ASPMail or ASPQMail (email is optional). As with ASPImage, many hosting services already have at least one (sometimes both) of these components installed. Check with your hosting service to find out which they have.

Installation Scripts

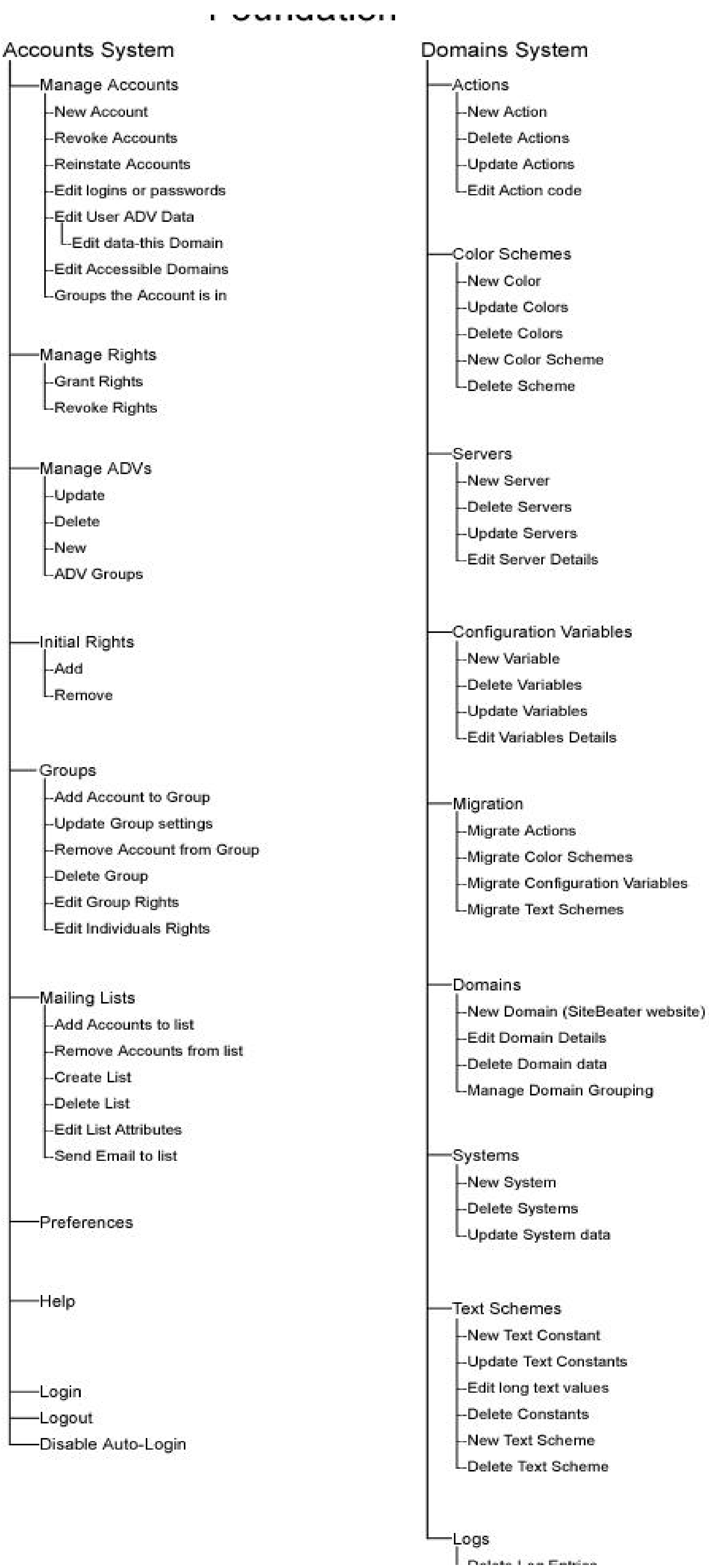
SiteBeater uses one installation script to install both the Domains and Accounts Systems (also known as the "Foundation"). Each other System has it's own installation script. These scripts all require that you know both the MSSQL server name and SA (or administrator) login and password. The scripts also require you to know path information for some of the files and website directories. You will probably *not* be given any of this information by your hosting service. In the case that your hosting company does not give you all of the required information, they will have to run the scripts for you.

The Foundation installation script also writes to a file on the server. Your hosting company will most likely have to set the permissions for you. The permissions must be set before the script is run, and can be reset after.

E

Navigation Quick Reference

See next page.


Accounts System
Manage Accounts
New Account
Revoke Accounts
Reinstate Accounts
Edit logins or passwords
Edit User ADV Data
Edit data-this Domain
Edit Accessible Domains
Groups the Account is in
Manage Rights
Grant Rights
Revoke Rights
Manage ADVs
Update
Delete
New
ADV Groups
Initial Rights
Add
Remove
Groups
Add Account to Group
Update Group settings
Remove Account from Group
Delete Group
Edit Group Rights
Edit Individuals Rights
Mailing Lists
Add Accounts to list
Remove Accounts from list
Create List
Delete List
Edit List Attributes
Send Email to list
Preferences
Help
Login
Logout
Disable Auto-Login
Domains System
Actions
New Action
Delete Actions
Update Actions
Edit Action code
Color Schemes
New Color
Update Colors
Delete Colors
New Color Scheme
Delete Scheme
Servers
New Server
Delete Servers
Update Servers
Edit Server Details
Configuration Variables
New Variable
Delete Variables
Update Variables
Edit Variables Details
Migration
Migrate Actions
Migrate Color Schemes
Migrate Configuration Variables
Migrate Text Schemes
Domains
New Domain (SiteBeater website)
Edit Domain Details
Delete Domain data
Manage Domain Grouping
Systems
New System
Delete Systems
Update System data
Text Schemes
New Text Constant
Update Text Constants
Edit long text values
Delete Constants
New Text Scheme
Delete Text Scheme
Logs

F

Glossary

Accounts System

- This is one of the two Systems that make up the Foundation. The Accounts System consists of all account information, polls, mailing lists, and account groups. Data for the Accounts System is stored in the SB_Accounts database.

Action

- Another term often used to describe a SiteBeater Action is a “macro”. Actions are more than just macros. Actions are the building blocks of a website. Actions can contain HTML and ASP. Being able to store HTML makes them perfect for page navigation. Being able to run ASP makes them ideal for small, encapsulated functions (aka “macros”).

  Every SiteBeater System uses Actions to open and close the pages within the System. Some Systems use Actions to display data – such as the polls or random image in the Image Gallery System.

Administrator

- Typically a user with rights to do many things is called an "Administrator". Because any user within the SiteBeater System can be assigned rights, any user can become an administrator.

ADV

- ADV is an acronym for Account Data Variable. ADVs are what store user information. For each piece of user information (first name, last name, email, etc.) there is an ADV. The ADV describes the user information. Each ADV has a description, type (either short or long text), optional right required to view it, and a System it appears in.

  Administrators may create new ADVs. Being able to create new ADVs allows site owners to gather additional and custom information about each of their users.

ADV Groups

- ADV data may be shared across Domains. To share ADV data from one Domain to another, those ADVs must be grouped together. Only one ADV from each Domain may be in an ADV Group.

Alternate View images

- In the Image Gallery System, Alternate View images are those additional images added to an image entry. Each image entry may have up to four Alternate View images. Users are required to have specific rights to view each of the four images. Alternate View images can be resided copied of the Primary Image, shots of the same image at various angles, or whatever other images the gallery owner wants to include in the image entry.

ALPHA
- One of the three server types. The ALPHA server is generally a development area for creating new Systems or updates to websites. The ALPHA server is usually not accessible to the general public.

ASPImage
- A third party component that allows SiteBeater to dynamically resize images and create thumbnails. This component can be purchased and downloaded from www.ServerObjects.com.

ASPMail
- A third party component that allows SiteBeater to send emails. This component can be purchased and downloaded from www.ServerObjects.com.

ASPQMail
- A third party component that allows SiteBeater to send emails. This is a more sophisticated version of ASPMail that allows the system to queue up mail to go out. This component can be purchased and downloaded from www.ServerObjects.com.

Auto-Login
- When a user logs in he or she can optionally choose to be automatically logged in the next time they visit the site. A cookie is set in their browser letting the system know that they want to be logged automatically the next time they come to the site.

BETA
- The second of the three server types. The BETA server is generally a test server. Data and code comes from the ALPHA server once it is ready for testing. Optionally the BETA server is usable by the general public or select people “off site”.

Board
- In the Message Board System. Boards and conferences are synonymous. Both contain threads of messages.

CDONTS
- This is CDO for NT – a Microsoft method of data manipulation. One of the features of CDONTS is the ability to route emails through an Exchange server. SiteBeater can use either CDONTS or ASPMail/ASPQMail to send its emails.

Color Scheme

- A Color Scheme is a set of Color Constants grouped together as a whole. The Aqua Color Scheme has many Color Constants within it that describe the Color Scheme Aqua. Color and Text Schemes are very similar in construct and usage.

Color Constant

- Color Constants make up Color Schemes. Color Constants consist of a name and value pair. The name is used in Action and ASP code to reference the value.

Conference

- In the Message Board System. Conferences and boards are synonymous. Both contain threads of messages.

Configuration Variable

- Configuration Variables store SiteBeater System settings.

Domain

- A Domain is a website entry in SiteBeater. Each website has a Domain entry that contains all the settings and information in the database for that website.

Domain Groups

- For new users to be able to log into multiple Domains, those Domains must be grouped together. That group is referred to as a Domain Group.

Domains System

- This is one of the two Systems that make up the Foundation. The Domains System consists of all Domains, Servers, Color Schemes, Text Schemes, Actions, and Configuration Variables. Data for the Domains System is stored in the SB_Domains database.

Foundation

- The Foundation is made up of the Accounts and Domains Systems. Every other SiteBeater System and website is made up using data stored in the Foundation. The Foundation also consists of the web pages that manipulate the data.

Gallery

- In the Image Gallery System. A gallery contains the individual image entries.

Home Domain

- A users Home Domain is the Domain on which the user initially created his or her account. The Home Domain is special because the user can always log onto their Home Domain. Also, certain administrative rights and functions affect users based on their Home Domain.

Image Entry

- In the Image Gallery System. An Image Entry is an entry in a gallery. Each Image Entry can have up to six images: the Primary Image, thumbnail, and up to four Alternate View images.

Initial Rights

- Rights that are given to all new users once they create their account. Initial Rights are not retroactive – meaning that after an account has been created, changes to Initial Rights do not affect that account.

LIVE

- One of the three types of servers. The LIVE server is sometimes called the "production" server. This is the server the general public uses.

Message

- In the Message Board System. A message is a users post to a board. The difference between threads and messages is that threads include the initial message and all replies. A message is only the individual post.

NT Authentication

- Microsoft's method of authenticating a login. The login must be created in the NT operating system. In a (compatible) browser, a popup window will appear asking the user for the login and password before they can access the page protected by the NT Authentication.

Primary Image

- In the Image Gallery System. In an image entry, the Primary Image is the first image uploaded when the image entry was originally created. The Primary Image is also the image displayed on the image detail page. Unlike Alternate View images, no special rights are required to view the Primary Image.

Rights

- Rights give users power to perform certain actions, see specific pieces of data, and manipulate that data. Rights also prevent unwanted users from doing these things. Users with rights are sometimes referred to as administrators.

Rotational Group

- A Rotational Group is an account Group whose accounts become valid only on certain days. Accounts within Rotational Groups are not meant to be long term.

Standard Group

- A Standard Group is an account Group whose accounts are normal, permanent accounts.

Text Scheme

- A Text Scheme is a set of Text Constants grouped together as a whole. The English Text Scheme has many Text Constants within it that make up the Text Scheme English. Color and Text Schemes are very similar in construct and usage.

Text Constant

- Text Constants make up Text Schemes. Text Constants consist of a name and value pair. Text Constants values can either be short (less than 255 characters) or long values. The name is used in Action and ASP code to reference the value.

Thread

- In the Message Board System. A thread is a series of messages and their replies.

TqcRunas

- A third party component that allows the SiteBeater System to read and write Domain data to and from other physical machines. This component may be purchased and downloaded from http://www.quimeras.com.

Users

- Users are visitors to a website. Users usually have no administrative rights. Once a user is given administrative rights, their designation changes to administrator. Users do sometimes have rights to do such things as create galleries or view Alternate View images.

SiteBeater