Caucus 3.1

Installation and Manager's Guide

Screen Porch LLC.

4031 University Drive, Suite 200
Fairfax, VA 22030-3409
http://screenporch.com

Revision Date: 18 February 1997

COPYRIGHT

TRADEMARKS

Chapter 1

Introduction

1.1 How to Use This Guide

This guide contains complete instructions for installing and managing the Caucus conferencing system on your Unix host computer. This manual is divided into several major sections:

- Chapter 1: Introduction
  General information about Caucus and this manual.

- Chapter 2: Installing Caucus
  How to install Caucus on a Unix host server computer.

- Chapter 3: Configuring Your Web Server
  How to configure your web server to work with Caucus.

- Chapter 4: Userids and Passwords
  How to handle userid and password security.

- Chapter 5: Managing Caucus
  How to create new conferences, delete old ones and, in general, manage Caucus on your computer.

- Chapter 6: The Caucus Architecture
  How Caucus works with the World Wide Web, and the location and purpose of the important files.

- Chapter 7: Customizing the Web Interface
  How to modify the Web interface for your site.

1.2 Purpose

Caucus is a software package that provides for computer conferencing, also known as computer-mediated or electronic conferencing. Computer conferencing is a way to enhance group communication. A computer conference is like a meeting except that participants need not be in the same place nor meet at the same time. Everything that everyone "says" is entered

into Caucus and stored in the memory of the host computer. Later participants review what others have entered and may add further comments. This process can continue indefinitely. Caucus keeps track of all the meeting details, such as who has read which discussions and what new material has been added.

Conferencing is used to supplement live meetings, to replace meetings where timing is not critical, or to bring together people who ordinarily would have a difficult time getting together due to location or cost constraints.

1.3 Interfaces

With Caucus 3.1, your conferencing users can choose from 3 different interfaces:

World-Wide-Web. Any HTML 2.0 compliant Web browser (with nested <TABLE> support) becomes an intuitive graphical user interface to Caucus conferences.

Menus. Terminal or telnet users with a login id on your host may choose a menu-driven interface that provides full access to conferences.

Command Line. A rich command language is available to power users. It also requires a login id on your host computer.

The installation procedure describes how to install all three interfaces on your host computer.

1.4 Required Background

This document assumes that you are familiar with normal system management functions associated with running a Unix host. It also requires that you be able to manage and configure an HTTP server on your host.

1.5 Security

A Caucus conference is as secure as the Unix operating system will allow. That is, when Caucus is properly installed, only the root and Caucus userids can directly access the files that make up a conference. Only the system manager should know the password to these userids.

The only way that other users can access a conference is by using the Caucus programs, which are installed with the "Set Userid" protection bit. The Caucus programs rigidly control who can enter or change text in a conference.

The World Wide Web interface uses the standard Web "access authorization" userid and password method to control access to the Caucus directories. The Web interface also uses several methods, such as client IP verification, to prevent "spoofing" by unauthorized users.

Individual Caucus conferences may be open to everyone or limited to certain people. For more information about conference security, see the conference “Organizer Guides” in the Documentation pages on Screen Porch’s web site, at http://screenporch.com.

Chapter 2

Installing Caucus

2.1 Before You Begin

The system manager must install Caucus. Ordinary users do not have the proper permissions to install Caucus.

If you are upgrading a Caucus site, we recommend that you have or make an adequate backup of the Caucus home directory. Then skip directly to section 2.3. Your Caucus software will be updated without harming any of your existing conferences. If you are installing Caucus for the first time, follow all of steps shown below.

The installation procedure installs the Caucus database, the World Wide Web interface, and the text interface. The normal Caucus 3.1 kit is licensed for a specific number of Web users, but limits the number of text-interface users to one at a time. If you have purchased the text-interface license option, you will also be allowed an unlimited number of text interface users.

(If you find that you need assistance with installing Caucus, start by joining our technical support conferences at http://screenporch.com.)

2.2 Create the Caucus Userid

Create a new userid, called "caucus", with its own home directory. (You may use a different name if you prefer. The installation procedure will adapt to whichever name you choose.) The home directory for Caucus must have enough free disk space to contain all of the Caucus programs and data files, and all of the anticipated conference data. A minimum of 100 megabytes is recommended. (The software itself is about 20 megabytes, maximum.)

Only the system manager, or a designated Caucus manager, should know the password to the Caucus userid.

2.3 Install the Caucus Software

The software installation is the same whether you are:

- installing Caucus for the first time
- upgrading your existing Caucus software

The installation procedure automatically determines if this is a new installation of Caucus or an upgrade to an existing Caucus site. If you are upgrading Caucus, your existing conferences will not be affected by the upgrade.

2.3.1 Login to the Caucus userid. (This is the userid that owns the Caucus files, from section 2.2.) You must actually log in to this userid; do ***not*** use "root" or "su" access.

These instructions in this guide assume that the Caucus home directory is in /home/caucus. In this guide, whenever you see the path **/home/caucus**, replace it with the actual pathname of the Caucus home directory on your system.

Important Note: unless otherwise stated, all commands must be typed while logged in as Caucus, and in the Caucus home directory.

2.3.2 Cease using Caucus. If you are upgrading Caucus, *all* Caucus users should exit or quit the program while you are performing the upgrade. To make absolutely certain of this, type the commands:

```
mv  BIN2/caucus_x  BIN2/caucus_x.old
./swebstop
```

2.3.3 Unpackage the software

The Caucus software is delivered in a file called **caucus31.t.Z**. In the Caucus home directory, type the command below to unpackage this file.

```
zcat caucus31.t.Z | tar xvf -
```

2.3.4 Run the installation script

The software includes an installation script that will automatically create the proper script files, set the proper file permissions, and so on. To run the script, type:

```
./cinstall
```

The script will ask for the hostname (and port number, if needed) of your web server. Be prepared to provide these.

2.3.5 Warnings

The cinstall script will produce some warnings and informative messages on your screen. A copy of these warnings is also placed in the file **caucus.warn**. These warnings should be self-explanatory, and fall into three categories:

- Information about specific files, paths, or URLs. For example, cinstall tells you the full URL for accessing Caucus.
- Warnings about new versions of old files. For example, cinstall creates a new SWEB/swebd.conf file, and renames your old swebd.conf file.
- Warnings about upward compatibility. For example, if you have conferences created prior to version 2.6 of Caucus, you will be warned to run the "fixdate" script on your conferences.

You should examine these warnings carefully and determine if any of them apply to your Caucus installation.

2.3.6 CV2 script for text interface

The cinstall script also creates two scripts called **cv2** and **cv2check**. Cv2 is the script used to run the text interface to Caucus. Cv2check provides a quick summary of how much new information there is in the conferences that you belong to.

If you have purchased the unlimited text-interface license option for Caucus, and are providing access to the text interface to your users, you probably want to copy these scripts to a public directory, such as /usr/local or /usr/local/bin. You may also wish to rename the scripts to something more mnemonic, perhaps **caucus** and **caucuscheck**.

2.3.7 Check hostname and port number

When you ran the cinstall script, it asked for a hostname and port number (such as "www.xyz.com" or "host.state_univ.edu:8001"). If you need to change this information, now or at any future time, edit the files listed below, and change the hostname or port number appropriately.

SWEB/swebd.conf
SWEB/start.cgi
public_html/caucus.html

2.3.8 Public HTML directory

Caucus places certain files in the Caucus userid's public HTML directory. The standard name for this directory is **public_html**. The Caucus distribution includes a public_html directory with the necessary files already in it. If your httpd server uses a different name, rename public_html to that directory name now.

For example, if your httpd server uses "WWW" as a user's public HTML directory, from the Caucus home directory type:

```
mv  public_html  WWW
```

You may also need to change the definition of the Caucus parameter "Caucus_Lib". See your SWEB/swebd.conf configuration file for details.

If for some reason your site does not support public HTML directories for individual userids, you can move the contents of **public_html** to another directory. In that case you will need to change the Caucus_Lib parameter in SWEB/swebd.conf, and the definition of the "img" variable in CML/SP31/Local/switch.i. See those files for details.

2.4 The Caucus Management Menu

Most of the common Caucus administration functions are provided in a menu in the shell script **manager_script**. To see this menu immediately on login to the caucus userid, run it directly from your .profile or .login file, as described below:

If you are running the Bourne, Bash, or Korn shell (sh, bash, or ksh), you will usually see a "$" as your Unix command prompt. Edit the file .profile, and at the very end, add the lines:

```
$HOME/manager_script
code=$?
if test "$code" = "1"; then
   exit
fi
```

Alternately, if you are running the C shell (csh or tcsh), you will usually see a "%" as your Unix command prompt. In that case, edit the file .login, and at the very end, add the lines:

```
$HOME/manager_script
if ( $status == 1 ) then
   logout
endif
trap 2
```

In either case, logout and then log back in again, and you should immediately see the Caucus management menu.

2.5 Starting and Using Caucus

2.5.1 Start the Caucus server

To start the Caucus server, choose the appropriate selection from the Caucus Management Menu.

You should also add the lines shown below to your system start-up file (such as /etc/rc.d/rc.local, or whatever it is called on your host) so that the Caucus server will start when your system reboots.

```
rm -f /home/caucus/SOCKET/sweb
/home/caucus/SWEB/swebd  /home/caucus/SWEB/swebd.conf
```

2.5.2 Using Caucus

The Caucus installation script creates a default HTML page for accessing the Caucus conferences on your host. It is located in **/home/caucus**/public_html/caucus.html. The URL for this file is:

http://**yourhost**/~**caucus**/caucus.html

This file is just a template for how to access Caucus from the Web. If your organization already has a set of web pages, you will probably want to integrate this file with your existing pages. You might choose to copy the links in this file to the appropriate places on your existing web pages; or you might decide to edit the caucus.html file and simply make it look more like your other web pages.

At this point you can look at this file in a browser, but the links from it will not become properly active until you finish configuring your web server, as described in the next chapter.

Chapter 3

Configuring Your Web Server

3.1 Web Servers

Caucus works with most standards-compliant httpd servers. This guide provides specific directions for:

1. the NCSA httpd server (http://hoohoo.ncsa.uiuc.edu/)
2. the Apache httpd server (http://www.apache.org)
3. the Netscape Enterprise Server 2.x (http://home.netscape.com)
4. the Netscape Communications Server 1.x (obsolete, but may still be in use)
5. the CERN httpd server (http://www.w3.org/pub/WWW/Daemon)

   (**Note**: while we include directions for the CERN server, we specifically recommend **against** using it. Our experience suggests it is too buggy for practical use with Caucus, and we do not guarantee support for sites using it.)

Caucus will most likely work with other httpd servers, but you must understand in detail how to define CGI files or directories, and how to set up access authorization controlled directories.

3.2 General Web Server Configuration Instructions

This section describes, in the abstract, the changes that must be made to your web server configuration to make it work properly with Caucus. Subsequent sections describe the precise details of these changes for the servers listed above.

Read this section to understand **what** you need to change; then skip to the section for your web server to see **how** to make those changes.

Once you have made these changes, your Caucus site will be up and running! But you should continue on and read chapter 4, "Userids and Passwords", to understand the implications of userids and passwords on the Web.

3.2.1 Define CGI directories

Caucus uses several different CGI programs in the directories SWEB and REG to communicate with the web server. The best way to identify these programs to the web server is to declare SWEB and REG as CGI directories.

Specifically, declare the following mappings of URLs to CGI directories:

http://yourhost.com/sweb/ to **/home/caucus**/SWEB/
and http://yourhost.com/reg/ to **/home/caucus**/REG/

If for some reason you cannot declare a CGI directory, enable your server in some other way to treat the files:

/home/caucus/SWEB/start.cgi
/home/caucus/SWEB/swebsock
/home/caucus/REG/swebsock

as CGI programs.

(Note: in all of the above, replace **/home/caucus** with the full pathname of the Caucus home directory.)

3.2.2 Define special "/caucus" URL

Caucus users who have already registered a userid may go directly to specific conferences, items, or responses through the special URLs shown below (instead of going through the regular caucus.html page in section 2.5.2).

http://yourhost.com/caucus	("Caucus Center" page)
http://yourhost.com/caucus/conference_name	(conference home page)
http://yourhost.com/caucus/conference_name/item	(go to that item)
http://yourhost.com/caucus/conference_name/item:response	(go to that response)

In order to make these special URLs work, the web server must be configured to map URLs that begin "http://yourhost.com/caucus" to the CGI file **/home/caucus**/SWEB/start.cgi. (This may not be possible for all sweb servers. Users of such servers can still access Caucus through the regular caucus.html page.)

3.2.3 Restrict Access with userids and passwords

Caucus' security requires that each user be identified by a unique userid and password. Caucus uses the standard web "access authorization" protocol to implement userid and password checking.

To enable access authorization for Caucus, you must declare that the directory **/home/caucus**/SWEB is protected by a userid and password database file. For some web servers, this is done automatically by the Caucus installation script.

If you have enabled the Caucus "file library" (see the Caucus_Lib parameter in **/home/caucus**/SWEB/swebd.conf), you may also wish to apply the access authorization protection to the file library directory. Without such protection, anyone on the Web can read or download files placed in the file library. With such protection, only Caucus users (with a userid and password) will be able to see those files.

The default location of the file library is **/home/caucus**/public_html/LIB. To protect this directory, do to it whatever you did to protect the **/home/caucus**/SWEB directory.

3.2.4 Restart your server

Some web servers must be shutdown and restarted before any of the previous changes will take effect. See your web server documentation for details.

3.3 NCSA or Apache Web Server Instructions

This section describes the precise details of configuring the NCSA or Apache web server to work with Caucus. It assumes that you have already installed your web server and are generally familiar with the server configuration files.

3.3.1 Define CGI directories

Find the httpd configuration file **srm.conf**. Edit it, and add the lines:

```
ScriptAlias  /sweb/   /home/caucus/SWEB
ScriptAlias  /reg/    /home/caucus/REG
```

(replacing **/home/caucus** with the home directory of the Caucus userid on your system).

3.3.2 Define special "/caucus" URLs

Also in srm.conf, add the lines:

```
ScriptAlias  /caucus/ /home/caucus/SWEB/start.cgi/
ScriptAlias  /caucus  /home/caucus/SWEB/start.cgi
```

3.3.3 Restrict Access with userids and passwords

Access authorization for NCSA and Apache servers is set up automatically by the Caucus installation script. It creates the file **/home/caucus**/SWEB/.htaccess, which declares that the directory is password-protected. That file points in turn to the userid and password database file **/home/caucus**/caucus_passwd, which is also set up by the Caucus installation script.

To add userids to the database file, use the shell script manager_script. Or your users may self-register a userid and password from the link in the caucus.html page (see section 2.5.2).

To also protect the Caucus file library, copy the file **/home/caucus**/SWEB/.htaccess to **/home/caucus**/public_html/LIB.

3.3.4 Restart your server

You must restart the httpd server in order for these changes to take effect. See your server documentation for details.

3.4 Netscape Enterprise Server 2.x Instructions

This section describes the precise details of configuring the Netscape Enterprise Server, version 2.x, to work with Caucus. It assumes that you have already installed your web server and are generally familiar with server configuration.

3.4.1 Define CGI directories

From the server configuration page, choose "Programs", sub-selection "CGI directory". Add entries for:

URL prefix "sweb/", CGI directory **/home/caucus**/SWEB/
URL prefix "reg/", CGI directory **/home/caucus**/REG/

And choose "save and apply" these changes (replacing **/home/caucus** with the home directory of the Caucus userid on your system).

3.4.2 Define special "/caucus" URLs

From the server configuration page, choose "Programs", sub-selection "CGI directory". Add an entry for:

URL prefix "caucus/", CGI directory **/home/caucus**/SWEB/start.cgi

And choose "save and apply" these changes.

3.4.3 Restrict Access with userids and passwords

In the directory **/home/caucus**/SWEB, create a world-readable file called **.nsconfig**, containing the lines:

```
<Files *>
RequireAuth userfile=/home/caucus/caucus_passwd realm=Caucus userpat=*
</Files>
```

From the server configuration page, chose "System Settings", sub-selection "Dynamic Configuration Files". In the "file name" field, type ".nsconfig". Do not change any of the other settings or checkboxes in this form. "Save and apply" these changes.

To add userids to the database file, use the shell script manager_script. (Do **not** use the Netscape server user database functions.) Or your users may self-register a userid and password from the link in the caucus.html page (see section 2.5.2).

To also protect the Caucus file library, copy the **.nsconfig** file you just created, to the directory **/home/caucus**/public_html/LIB.

3.4.4 Restart your server

It is not necessary to restart the Netscape Enterprise server. All of the changes will take effect immediately.

You may, however, have to explicitly enable “user document directories” (i.e., the public HTML directories) in order for Caucus to work. To do this, from the server configuration page, choose “Content Management”, sub-selection “User Document Directories”. You don’t need to make any changes, just click on “OK” and “Save and Apply” to enable the user document directories.

3.5 Netscape Communications Server 1.x Instructions

This section describes the precise details of configuring the Netscape Communications Server, version 1.x, to work with Caucus. (This server is now obsolete, but may still be in use in many locations.) It assumes that you have already installed your web server and are generally familiar with server configuration.

In all of the instances below, replace /home/caucus with the full pathname of the Caucus home directory.

3.5.1 Define CGI directories

From the Netscape server manager, under the section **CGI and Server Parsed HTML**, choose Specify a directory that will contain CGI programs only.

In the **URL prefix** box, enter "sweb". Then in the **CGI directory** box, enter "**/home/caucus**/SWEB". Make those changes.

Repeat the same process for the **URL prefix** "reg", **CGI directory** "**/home/caucus**/REG".

3.5.2 Define special "/caucus" URLs

Repeat the process from 3.5.1 for the **URL prefix** "caucus", **CGI directory** "**/home/caucus**/SWEB/start.cgi".

3.5.3 Restrict Access with userids and passwords

To ensure full compatibility with Caucus, you must manually create a Netscape user database sub-directory that is owned by the Caucus userid. Assuming that your Netscape server is installed in **/var/ns-home**, and that the Caucus userid is called **caucus**, type the commands below. (Note that you must know the "root" password in order to do this.)

```
su -
cd /var/ns-home/userdb
mkdir caucus
cp /home/caucus/caucus_passwd caucus/passwd.pwf
chown -R caucus caucus
exit
```

If you wish to allow your users to change their own passwords or self-register their own userids, you must also manually edit the files passprog, webreg, and manager_script in the Caucus home directory. In those files, replace

/home/caucus/caucus_passwd
with **/var/ns-home**/userdb/caucus/passwd.pwf

From the Netscape server manager, under the section **Access Control and Dynamic Configuration**, choose Restrict access to part of your server through authentication.

The **Restrict Access** page should say "You are currently modifying the directory **/home/caucus**/SWEB/*". If it says "You are currently modifying the entire server", then choose Browse Files. In the **Choose a directory to list from** box, enter "**/home/caucus**/SWEB". Select the radio buttons for **Only list directories**, and **Follow sym-links**. Make those changes. If there is a link for **Choose this directory**, select it.

Return to the **Restrict Access** page. It should say that "You are currently modifying the directory **/home/caucus**/SWEB/*".

In the **Which Database?** box, select or type "caucus/passwd". Leave the **Which users?** box blank (unless you have a reason to restrict Caucus to specific set of users).

In the **Realm** box, type "Caucus". Make these changes.

To add userids to the database file, use the shell script manager_script, or the **Database user manipulation** section of the Netscape server manager pager. Or your users may self-register a userid and password from the link in the caucus.html page (see section 2.5.2).

3.5.4 Restart your web server

Back at the server manager page, under **Server Control**, choose Start up, restart, or shutdown your server, and press the **Restart!** button.

3.6 CERN Web Server Instructions

This section describes some of the details of configuring the CERN web server to work with Caucus. It assumes that you have already installed your web server and are generally familiar with server configuration. (**Note**: Screen Porch recommends **against** the use of the CERN web server. This information is provided here purely for those who wish to, or have some need to, experiment with the CERN server.)

3.6.1 Define CGI directories

Find your httpd configuration file, typically **httpd.conf**. Edit it, and add the lines:

```
Exec  /sweb/*     /home/caucus/SWEB/*
Exec  /reg/*      /home/caucus/REG/*
```

(replacing **/home/caucus** with the home directory of the Caucus userid on your system).

3.6.2 Define special "/caucus" URLs

In **httpd.conf**, add the line:

```
Exec  /caucus/*  /home/caucus/SWEB/start.cgi/*
```

3.6.3 Restrict Access with userids and passwords

In httpd.conf, add the lines below:

```
Protection PROT-SETUP-USERS {
    UserId       nobody
    GroupId      nogroup
    ServerId     caucus
    AuthType     Basic
    PasswdFile   /home/caucus/caucus_passwd
    GroupFile    /home/caucus/groups
    GET-Mask     users
}
Protect  /sweb/*     PROT-SETUP-USERS
```

3.6.4 Restart your web server.

You must restart the httpd server in order for these changes to take effect. See your server documentation for details.

3.6.5 Replace swebsock program with a shell script

Due to idiosnycracies of the CERN httpd server, it is necessary to replace the 'swebsock' program with a shell script which invokes the actual program. To do this, go to the directory **/home/caucus**/SWEB, and type:

```
mv swebsock  swebsock2
chmod 4711   swebsock2
echo "#!/bin/sh"                          >swebsock
echo "exec /home/caucus/SWEB/swebsock2" >>swebsock
chmod 755 swebsock
```

3.7 If Something Goes Wrong...

The instructions in the previous sections should guide you through a successful installation of Caucus. If, however, you find that Caucus does not appear to be working, here are some common possible problems and their solutions.

(In checking these possibilities, be very careful about spelling and capitalization. In the Unix world, upper and lower case file names refer to different files.)

Note: if you make a change to correct a problem, you may also need to restart your httpd server before the changes take effect.

3.7.1 Browser cannot find http://yourhost/~caucus/caucus.html

Check the file permissions on caucus.html. It must be readable by your httpd server. Typically that means it must be world-readable, but that depends on your particular site.

Check to ensure that your web server has been enabled to serve individual users' **public_html** directories. Some web servers require that this option be explicitly turned on.

If your web server uses some name other than public_html for individual users' web directories, make sure that you rename /home/caucus/public_html (see section 2.3.8).

3.7.2 The Caucus Center Page link produces a server error.

Make sure your Caucus server ("swebd") is running. See section 2.5.1.

Check the ScriptAlias (NCSA httpd srm.conf), Exec (CERN httpd configuration file), or CGI directory (Netscape CGI and Server Parsed HTML) settings for your httpd server, section 3.2.1 etc.. **Restart** your httpd server. If the problem continues, it is likely that you have not properly configured your CGI directories.

Check all of the files listed in section 2.3.7. Make sure they have the proper hostname and port number.

Make sure your httpd password file exists (see sections3.3.3, 3.4.3, etc.). If, for example, you see an httpd password file called "caucus_passwd.OLD", but no corresponding file "caucus_passwd", copy the "OLD" file to the original file.

Check the ownership and permissions on the programs in **/home/caucus**/SWEB. The files swebd and swebsock should be owned by the caucus userid, and should have permission masks of 4711 ("-rws- -x- -x"). The CML files in **/home/caucus**/CML/SP31 should be owned by and readable by the caucus userid. Verify that the CML file named in the URL in your browser's location window actually exists.

3.7.3 Caucus does not know who you are

If Caucus does not know who you are, does not appear to know your userid, or thinks that everyone is the same userid, then your web password file is probably not in place.

Start by exiting your browser and starting it again. When you try to go to the Caucus Welcome page, you should be prompted (by your browser) for a userid and password.

For NCSA httpd, check the caucus_password and SWEB/.htaccess files. Make sure they are readable by httpd. Restart your httpd server.

For the Netscape Communications server, make sure that **/home/caucus**/SWEB/* has "restricted access through user authentication", and that it is using the database "caucus/passwd.pwf".

For the Netscape Enterprise server, make sure that access control has been enabled.

3.7.4 Other Problems?

Please join our technical support conference at http://screenporch.com. Updated releases of this installation guide are available directly from our web site. If you have an interesting or unusual problem, we want to hear about it and include it in our documentation.

Chapter 4

Userids and Passwords

4.1 Introduction

Each person who accesses Caucus from the Web needs a userid and a password. These userid and password pairs are stored, encrypted, in a special password database file that is used by the httpd server. (Note that this database file is ***not*** the same as the regular Unix password files /etc/passwd or /etc/shadow file, although it does have a similar format.)

There are many different ways to set up and control the password database. The Caucus installation procedure sets up the password database in one particular way that is believed to be the most common and most useful.

Depending on the purpose of your site, and how your users access your host, you may wish to change how userids and passwords are controlled and stored. The rest of this chapter describes some of the other methods for handling userids and passwords, and their implications.

4.2 Default Method: Pure Web Access

The default method (as set up by the Caucus installation procedure) is called "pure web access". This model assumes that your host is used only for providing Web services. The only access your users have is (a) reading pages that are published on your server, and (b) using Caucus conferences. This is the typical situation for an existing Web server that is simply adding conferencing capabilities.

4.2.1 caucus_passwd file

In this method, your users' access to Caucus is controlled by the NCSA (or "flat") style password file **/home/caucus**/caucus_passwd. The contents of this file are modified by three different tools or features:

1. New "would-be" Caucus users may self-register a new userid and password (from the caucus.html page in section 2.5.2).

2. Individual Caucus users may change their own passwords (by clicking on their own name and filling in the appropriate form).

3. The Caucus Management Menu (in the shell script manager_script) may be used to add userids to the file, or change any user's password.

4.2.2 Configurable options

In the default method, there are also three configurable options:

1. You may disable the self-registration of new Caucus userids (in which case the Caucus Manager must create them). To disable self-registration, backup the file **/home/caucus**/public_html/caucus.html, and then edit it and remove the "self-register" link.

   Then rename the directory **/home/caucus**/REG to **/home/caucus**/REG.off.

2. You may disable the ability of individual users to change their own passwords, by editing **/home/caucus**/CML/SP31/Local/switch.i, and setting "change_password" to 0.

3. You may require that all users confirm their password **each** time they enter Caucus. This is a good idea if your users will be accessing Caucus from public workstations. Most Web browsers remember ("cache") userids and passwords, which makes it entirely too easy for a user at a public station to impersonate the immediately preceeding user at the same station.

   To require password confirmation, edit **/home/caucus**/CML/SP31/Local/switch.i, and set "confirm_password" to 1.

4.3 Mixed Method: Web and Unix Access

Another common model is "mixed" access to both Web and Unix userids. In this model, the same group of users have both Unix login access and Web access to your host. They may use the Unix access for e-mail, or to run the Caucus text-interface, or for a variety of other purposes, but in any case the point is the same. They should have (and will expect) to use the same userid and password, whether they are logging in to Unix or accessing Caucus conferences from the Web.

This scenario is more likely to apply to an organization (such as a university) that is already providing services (such as e-mail) to its users, and is now adding conferencing capabilities.

In this case, the Web userid and password file must derive from (be copied from) the Unix system password file(s). Caucus includes a utility to perform this function, called wpwcopy. You may use this utility as provided, or modify the source code (provided in the file wpwcopy.c) as needed for your own site.

The simplest way to do this password copying is to make it happen automatically every day. Then as new users are added to the Unix system password file(s), they will also get added to the Web password file.

The directions below show how to make this occur automatically every day at 3 a.m. They assume that all of your user passwords are kept in /etc/shadow or /etc/passwd, and that you are familiar with the managment of your Unix system. (If you are using a different password database, you must adapt the wpwcopy program to read from that database, or find some other way to export your userids to an NCSA-style password file.)

If your site stores encrypted user passwords in /etc/shadow, you can either

(a) permit wpwcopy as setuid 'root', and run it from crontab as Caucus, or
(b) run wpwcopy directly from root's crontab.

In either case the crontab entry looks like:

```
0 3 * * * /home/caucus/SWEB/wpwcopy  \
          /home/caucus/caucus_passwd \
                      /etc/passwd  /etc/shadow
```

Some systems may only use /etc/passwd (and not /etc/shadow). This is pretty insecure (and mildly dangerous), but if that's what you're using, the crontab entry should be:

```
0 3 * * * /home/caucus/SWEB/wpwcopy  \
          /home/caucus/caucus_passwd \
                      /etc/passwd
```

If you are running the Netscape Communications Server version 1.x, change the above crontab entry(ies) to reference:

```
          /var/ns-home/userdb/caucus/passwd.pwf
(instead of)  /home/caucus/caucus_passwd
```

The wpwcopy program provides additional options, such as the capability to copy only a restricted range of userids. See the comments in wpwcopy.c for more information.

Because userids and passwords derive from the Unix system password file(s), Web users in this scenario cannot self-register, or change their passwords. You should disable these features to prevent confusion. See section 4.2.2 for details.

If you are using /etc/shadow password protection, and are concerned about the possibility of someone reading the encrypted user passwords from the Web password file, you can add a level of protection to the standard Web security. (A detailed description of this process is beyond the scope of this guide, but the system manager should be able to implement it from the description below.)

This protection involves making the Caucus web password file readable by your httpd server, and no-one else. The simplest approach is to define a new group, and make your httpd server setgid to that group. Then change the group ownership of the web password file to the new group. Do not use the new group for any other purpose.

4.4 Other Methods

Many other methods are possible. Several are described below, to spark your thinking. Screen Porch does not directly provide or support these scenarios as of this writing, but there is also no reason why they would not work.

1. **Combined pool of Unix users and Web-only users**. This is a hybrid of the first two methods. It requires defining a file that contains userids and passwords of Web-only users. This file could be modifiable by the Caucus webreg and passprog programs, i.e. Web users could self-register and change their own passwords. Then, on a regular basis (such as provided by crontab), combine the Unix system password file(s) with this new file, to produce the "real" Web password file.

2. **Allow Web users to change their Unix and Web passwords**. This requires writing a CGI script that can run the Unix 'passwd' (or equivalent) utility. It opens a number of possible security concerns, but in theory it could be done safely.

3. **Use DBM files for Unix and Web access control**. The instructions for the first two methods use text or "flat" password files. Both scenarios could use DBM files instead, which are faster for large numbers of userids. The wpwcopy program would have to be modified to handle DBM files instead of flat files, or a new program written to replace it.

4. **Use "captive" Caucus for Unix access.** The text interface for Caucus provides an option for a special mode, called "captive", where only a single Unix userid is needed. This userid runs Caucus (and only Caucus -- hence the "captive" name.). In this mode, Caucus manages a set of psuedo-userids and passwords for all the different users sharing this single Unix userid.

   To run Caucus in this mode, change the options file "captive.opt" to enable the "captive" and "webpasswd" options. (See the text interface guide Customizing the Caucus User Interface for more information on options files.) The text and the web interfaces will share the same set of userids and passwords, taken from the web password file.

   The advantage of this approach is simplicity. The disadvantage is that is that it really has nothing to do with Unix userids; it just provides a way to use the text interface. "Captive" users have no e-mail or other Unix access.

Chapter 5

Managing Caucus

5.1 Caucus Management Overview

Once Caucus is installed, it also needs to be managed. One person should be designated as the Caucus manager to handle the infrequent duties listed below. This person need not be a system manager, but does need to know the password to the Caucus userid.

Management of the individual conferences is done by the conference organizer(s). See chapter 12 of the Caucus User's Guide.

The Caucus manager's duties include:

- creating new conferences
- deleting old conferences
- optional preregistration of new users
- deleting users

In Caucus 3.1, these management duties must be performed by logging in to the Caucus userid, and using the Caucus Management Menu. Later releases of Caucus will include a Web interface to these management tasks.

5.2 Creating New Conferences

The Caucus software distribution includes a conference called **demonstration** that includes sample items from other Caucus sites, which demonstrate some of the uses of conferencing.

While this conference is a good way to see what Caucus is all about, it is recommended that you create a new, empty conference for all of the users on your system. If you later find that you need multiple conferences to serve distinct groups of people, you can create them as you need them.

To start a new conference, login to the Caucus userid and choose "Create a new conference".

You will be prompted for the name of the conference, the type of the conference (ignore the file library choice and select "conference"), and the userid of the organizer of the conference. The management program then will proceed to create the new conference.

The organizer of the newly created conference should join the new conference as soon as is convenient. See the chapter "Organizing a Conference" in the Caucus User's Guide.

5.3 Deleting Old Conferences

The time may come when you want to remove a conference from your computer. This can happen for several different reasons:

- Some conferences may be set up for a specific purpose or to plan for a specific event. Once the event is over, there may no longer be a need for that particular conference.

- A very active conference can get so big (several thousand items) that keeping track of everything becomes very difficult for its participants. In that case, the organizer may re-start the conference: that is, start a completely new, empty conference, copy over the most important items, and go on with a fresh slate. Once all the participants have switched over to the new conference, the old one can be removed.

If you re-start a conference, it is a good idea to save the contents of the conference on some backup medium before removing the old conference. There are two ways to do this.

1. **Save the actual conference data files**. Start by determining the conference number, with the command:

   ```
   grep conference_name MISC/confs*
   ```

   The conference number is the 3-digit number between the "#" signs. Then archive the directory C**nnn**, where **nnn** is that 3-digit number.

   This is a good method to choose if you may wish to restore the conference at some later date.

2. **Save the text of the conference**. Run the Caucus text interface (the "cv2" script). Join the relevant conference. If you see the "Standard Menu" banner, switch to "Standard Menus -- Advanced Form" with the M (menu) option. Type the command:

   ```
   show items all >file
   ```

   and archive **file**. This is a good method to chose if you just want to have a copy of the text of the conference available (to print, or to extract text for another purpose).

Finally, to delete a conference from the system, login to the Caucus userid and choose "Delete an old conference".

You will see a list of all of the conferences and be prompted for which one you want to delete. Caucus will ask you for confirmation before it deletes the conference you selected. To return to the menu, press <RETURN> when asked "Delete which Conference?".

5.4 Preregistering Users

Typically most Caucus users will register themselves. Once they have a userid and a password, the first time they use Caucus they will be prompted to register their name, e-mail address, a brief personal introduction, and so forth.

There is also a mechanism for pre-registering users in batches. This is really only advantageous if you have a very large group of people, or if there is a need to not have the users register themselves. For more information, see Appendix A.

5.5 Deleting Users

On occasion you may also wish to delete a user or users. For example, if you are managing Caucus in a large company, you might delete users who have retired, or have moved on to other organizations. Deleting a user removes the user from all of the conferences and frees the disk space taken up by the user's message and participation files. It does not remove any of the user's items or responses.

Note that deleting a user is not the same as removing a user from a particular conference. For more information about managing an individual conference, see the Caucus “Organizer Guides” on the Documentation pages of the Screen Porch web site, at http://screenporch.com.

5.5.1 Deleting a single user

To delete a single user, login to the Caucus userid and choose "Delete a person".

You will be prompted for the name of the user to be deleted. Caucus will ask for confirmation before it deletes the user. To return to the menu, press <RETURN> when asked "Kill which person?".

5.5.2 Deleting multiple users

You may delete groups of users that have not accessed Caucus since a certain date.

The **listuser** script will prompt you for a date. (Any date format, such "1/1/96" or "1-jan-96" will work). It will display all of the users who have not accessed Caucus since that date.

The **expuser** script works the same way, but it actually deletes the users.

Chapter 6

Caucus Architecture

6.1 Introduction

This chapter describes the overall architecture of Caucus 3.1. This includes the mechanics of how the web interface actually works, and the location, names, and formats of the most important files.

While it is not necessary – or even helpful – to read or understand this chapter in order to *use* or even to *install* Caucus, it is very helpful if you intend to modify the web interface, or to connect other applications or programs to Caucus.

This chapter assumes a general familiarity with HTML, Web server management, Unix commands and processes, and the use of Caucus.

6.2 Caucus 3.x Design Goals

There were seven main design goals that shaped the architecture of Caucus 3.x. Previous versions of Caucus provided a text and menu interface to the conferencing environment; version 3 is a completely new product, which adds web capabilities. Its design goals included:

1. Allow the use of *any* Web browser to provide a graphical user interface to Caucus conferences. (In practice, this has come to mean any graphical browser that supports the HTML <TABLE> tag.)

2. Provide the tools for Webmasters to build a completely customizable Caucus interface. Caucus 3.1 uses "CML" (Caucus Mark-up Language) scripts, which are analogous to individual HTML pages.

   Caucus includes a default set of such scripts (pages), but they may be completely customized by the local site. This is in keeping with the long-standing Caucus tradition of complete customizability.

3. The Caucus 3.1 server was built on top of the existing Caucus API (applications programmer interface) function library, minimizing development time and guaranteeing data compatibility.

4. Caucus 3.1 works side by side with existing Caucus ("text interface") software. A Caucus user may access conferences through the Web *or* the text interface, without conflict.

5. Caucus 3.1 works with existing Unix HTTP servers, through the CGI interface. The Caucus server could also be adapted to work with a custom HTTP server to provide for higher efficiency.

6. The Web "access authorization" userid and password scheme is used to provide secure access to Caucus. When a userid has been verified by the Web server, that same userid is used to identify the particular Caucus user. All normal Caucus security (access to specific conferences, etc.) applies.

7. Transactions between the browser and the Caucus server must be as efficient as possible. The main effect of this on the design is the creation of a dedicated "sub-server" process for each user's Caucus session.

6.3 Caucus Web Interface: Transactions

This section describes what actually happens when a person uses a Web browser to access Caucus. In the steps listed below, "swebd" refers to the Caucus server process. "Swebsock" is a light-weight program that passes data to and from swebd. "Httpd" is the standard name for the HTTP server process.

6.3.1 Initial connection to Caucus

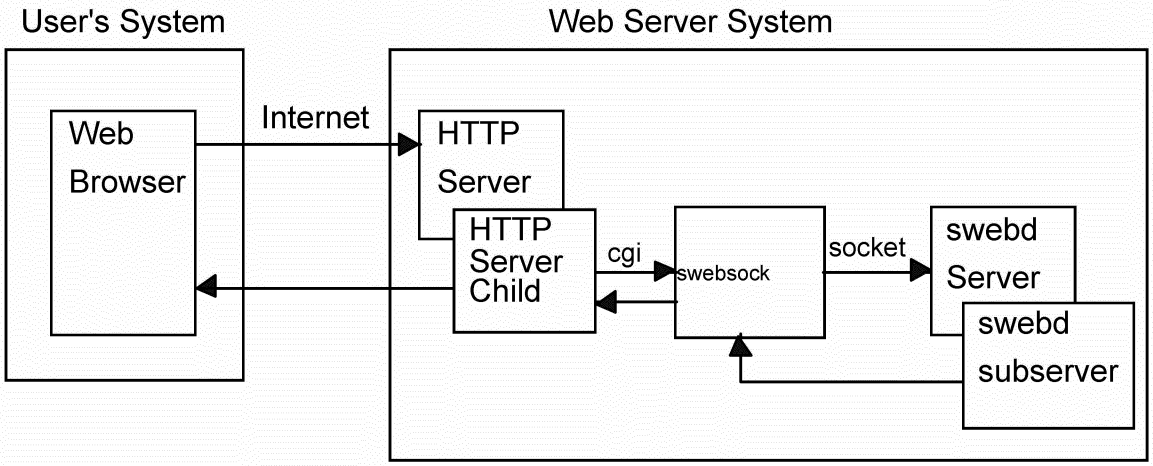
1. The user's browser sends a connection request (over the Internet, or a local intranet) to the host's HTTP server.

2. The HTTP server immediately spawns (or connects to a pre-existing) child httpd process to handle the request.

3. The initial "connection to the Caucus Web server" is actually an access-authorization (i.e., userid and password) protected URL that runs a CGI program called swebsock. Swebsock opens a socket to Swebd (the master Caucus server started in section 2.5.1).

4. Swebd spawns a child, called the swebd subserver, which reads the userid from swebcli. The subserver is now "dedicated" to this userid, and continues running on its own. The subserver passes its process id (and a unique security code) back to swebsock. Swebsock constructs an HTML page containing the process id and code, and passes this back through the HTTP server child to the browser.

This process is illustrated in the following diagram:



6.3.2 Subsequent requests

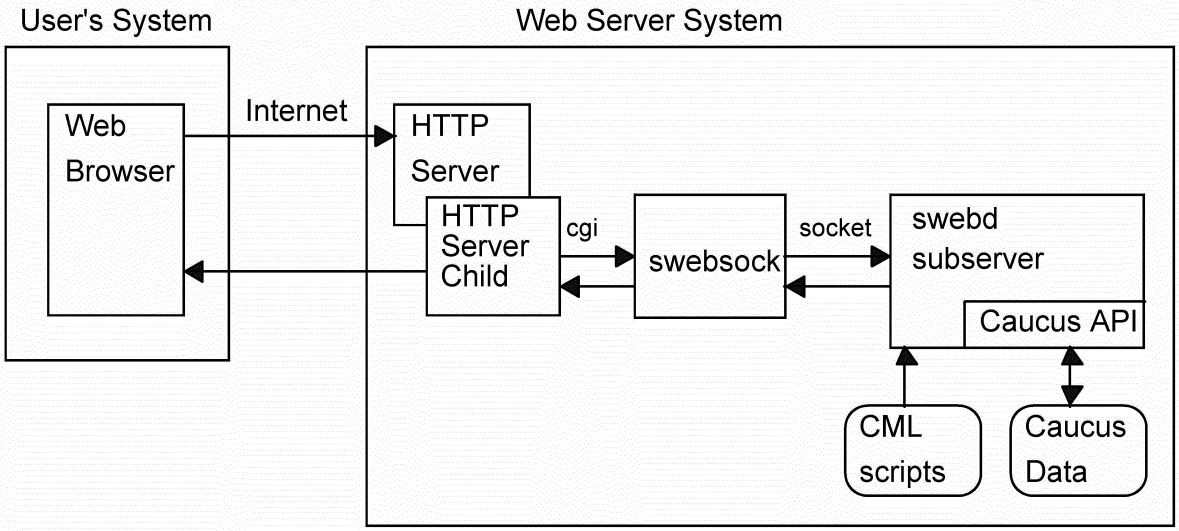
Once the initial connection is made, all subsequent Caucus requests by the browser are passed through to the dedicated swebd subserver. Each such request uses a particular CML script as part of the URL. Such a request will produce the following sequence of events:

1. The browser sends the new request to the HTTP server.

2. The HTTP server immediately spawns a child httpd to handle the request.

3. The URL runs a new instance of swebsock, which passes the request on to the dedicated subserver. The subserver reads (or writes) the requested information to the Caucus database, through the Caucus API. The subserver then formats the information according to the codes in the requested CML page, and passes the resulting dynamically created HTML page back through the HTTP server child to the browser.

This process is illustrated in the following diagram:



6.3.3 Notes

1. In the diagrams, the large dashed boxes are computer systems. The small boxes are processes, and the rounded boxes are disk files. Lines indicate communication paths, either HTTP, CGI (stdin/stdout), sockets, or file reading and writing.

2. In practice, the initial connection and the first subsequent request are combined in a single execution of swebsock, so that the initial connection to the swebd subserver returns an HTML page with real data in it. (Typically the "Caucus Center" page.) The overall model is the same, in any case.

3. Note that *each* browser request requires at least two new processes: the HTTP child, and the CGI swebsock. These processes are kept as lightweight as possible.

4. In contrast, since there is one swebd subserver per user, and each subserver persists across the entire user's browser session, the subservers cache all sorts of information. The subserver also has a timeout period -- i.e., after a certain period with no requests, it exits. Otherwise the system fills up with inactive subservers.

6.4 CML: The Caucus Markup Language

6.4.1 CML Description

The Caucus Mark-up Language is a superset of HTML. More precisely, HTML is embedded in CML scripts; Caucus does not actually understand or parse the HTML codes. CML scripts contain 4 kinds of text:

1. **Comments**. In the Unix tradition, all lines beginning with "#" are comments and are ignored. Entirely blank lines are also ignored.

2. **HTML code**. All lines beginning with a single quote (') are parsed for CML functions, but are otherwise passed on to the browser unchanged. (The quote is removed.)

3. **CML functions**. Strings of the form "$xyz()", "$xyz(value)", or "$(value)" are parsed and replaced by the appropriate Caucus values. The CML functions are described in the CML Reference Guide.

4. **CML directives**. Directives are like C program code: they describe actions to be taken. Directives include conditional statements ("if" and "else") and loop controls ("for" and "count").

A single logical line may be broken across several physical lines; a "\" as the last character means "continued on next (physical) line". This is generally unneeded, except for HTML <PRE> text that is being built out of mixed text and CML functions.

6.4.2 CML directives

The CML directives provide some simple control structures recognizable from other programming languages, including:

```
for variable in list
count variable first_val last_val
if condition
else
set variable value
```

For more information, see the CML Reference Guide.

6.4.3 CML functions

All CML functions evaluate to strings of characters. There is no other data type. The same holds true for CML variables. The CML functions provide access to Caucus data, browser and server control, string manipulation, and logic functions. Again, see the reference guide.

6.5 Layout of Caucus files

This section describes the layout of the Caucus files -- their location and purpose. All of the files live in or under the Caucus home directory, and (unless explicitly noted elsewhere) should always be owned by the Caucus userid.

Important: If you are editing these files for any purpose, you must do it while logged in as the Caucus userid. In particular, do ***not*** modify the Caucus files, or run the conference management programs, while logged in as "root".

6.5.1 CML pages

The CML pages control the precise look and feel of the Caucus 3.1 web interface. They are all located under the CML directory. As a site may have multiple (distinct) interfaces, each interface gets its own sub-directory under CML. The default set of CML pages is contained in the directory **CML/SP31** (SP for "Screen Porch").

The CML pages are ordinary ascii text files, usually called something.cml, or something.i (for "include" -- files included in other .cml files). Each CML interface (such as **CML/SP31**) also has a special subdirectory called **"Local"**. This contains files that are intended to be changed for your local site, and that will not be touched or replaced the next time you install a Caucus upgrade.

Two particularly important files in the Local subdirectory are:

switch.i contains common "switches" that may be set for your site to change how Caucus behaves.

l_confs.i list of conference names that will appear under "Popular Conferences" on the Caucus Welcome page.

See the header comments in these files for more information. See chapter 7 for more information about customizing the Web interface.

In addition to the conferencing interface in **CML/SP31**, there is also a separate (and small) interface in **CML/REG31**. This set of CML pages is entirely dedicated to registering a userid and password for a new user. (It must be a separate interface, because it will be used by people who have not yet gotten or been assigned a userid and password!)

6.5.2 The SWEB CGI directory

The SWEB directory contains CGI programs and related files that are used to start up the regular Web interface to Caucus. This includes

swebd	the Caucus server program
swebd.conf	the configuration file for swebd
swebsock	CGI program to communicate between httpd and swebd
.htaccess	file that makes **SWEB** an access-controlled directory (NCSA httpd)
cpw1	program to modify httpd password file
start.cgi	CGI script used to interpret "special" Caucus URLs, such as "http://hostname/caucus/conference_name/item_number.

6.5.3 The REG CGI directory

The REG directory contains CGI programs and related files that are used to start up the "register a userid" interface. This includes:

swebsock	a Unix “link” to the SWEB/swebsock program

6.5.4 The SOCKET directory

The various Caucus 3.1 programs (swebd, swebcli, swebt.cgi) communicate with each other via a Unix concept called "sockets". The sockets must have a name and a location; therefore they are placed in this directory. Files include:

sweb	socket to master swebd server
swebd**nnnnnn**	socket for a particular swebd subserver, process number **nnnnnn**
debug	if this file exists, debugging logs are created for **swebd** and **swebsock.**
sweblog.**nnnnnn**	debugging log for swebd process **nnnnnn**
talklog.**nnnnnn**	debugging log for swebsock processes that communicate with swebd process **nnnnnn**

6.5.4 The public_html directory

A URL of the form "http://yourhost/~caucus/xyz.html" looks for the file **xyz.html** in the **public_html** directory. (Depending on your httpd server, you may have renamed **public_html** to something else in section 2.3.8.) Caucus keeps some specific files in this directory:

caucus.html	Simple HTML page to link to Caucus interface (via **SWEB/caucus.cgi**) and to the "register a userid" interface (via **REG/register.cgi**).
GIF31	directory containing gif and jpeg images used by Caucus interface.

6.5.5 The BIN2 program directory

BIN2 contains all of the programs used by the Caucus text interface. These include:

caucus_x	The main Caucus 2.7 text interface program (run from the "**cv2**" script)
cauchk_x	The Caucus "check" program, run from the **cv2check** script.
caumnt_x	The Caucus maintenance program, run by the various management scripts (**cv2start**, **cv2remov**, **cv2kill**, etc.)

6.5.6 The DIC2 dictionary directory

DIC2 contains the source files for the Caucus text interface "dictionary". The text interface is completely customizable, and one site may host many different such interfaces. See Customizing the Caucus 2.7 Interface Guide for more information.

6.5.7 The GROUPS group permissions directory

Users may be given permission to access specific conferences by individual userid, or by groups of userids. These groups are defined in files in the GROUPS directory. For more information, see the Caucus "Organizer's Guides" at http://screenporch.com.

6.5.8 The Cnnn conference directories

The conference data for a particular conference is stored in a single directory. Each conference has a unique three digit number; thus, the data for conference number 1 is stored in the directory C001.

Conference data is always stored in "flat" ascii text files. In theory this means that the Caucus manager may edit these files directory. In practice you should never do this without specific instructions from Screen Porch technical support staff. This information is provided purely for reference; Screen Porch is not responsible for the results of unauthorized tinkering with these files.

Important files include:

userlist	permissions list of who may or may not access this conference
masteres	master list of items and number of responses to each item
0010000000	text of item 1 (and some responses)
0050210000	text of item 5, response 21 (and some following responses)
introduc	conference "introduction"
greet	conference "greeting"
membr001	list of conference members
variable	conference variables (from CML $set_conf_var() function)

6.5.9 The MISC (miscellaneous) Caucus-wide data directory

MISC contains files that relate to the entire Caucus site, not just a specific conference. As in section 6.5.8, these files should not be tampered with without specific instruction from Screen Porch. Important files include:

confs001	a list of conference names and their equivalent three digit numbers
dicti000	compiled version of text-interface "dictionary" number 0
names**nnn**	list of words in names of registered users, with mapping to their userid
bugslist	a log of possible Caucus "bug" conditions encountered on this host

6.5.10 USER001, Caucus user files

In addition to the conference-specific files, and the Caucus-wide data files, there is also data stored about each user. Data files for a userid **alpha** are stored under **USER001/alpha**. (Some Unix systems enforce so-called "sanity limits" on the number of sub-directories in a directory; if your system is one of them, Caucus may automatically create directories USER002, USER003, and so forth as needed.)

Important files include:

register	"Registration" information about this person, including their name, telephone number, brief self-description (introduction), and so forth.
p0010000	participation record in conference 1
variable	user variables (from the CML function $set_user_var().)

6.5.11 TEXT001, temporary user files

Temporary files created for each user (for example, during the entry or editing of items and responses) are stored here. It has the same structure as **USER001** (one sub-directory per userid). **Note:** The permissions for this directory and its sub-directories should be write-all.

6.5.12 Files in the Caucus home directory

There are some Caucus files which do not fit in the purposes described for the previously listed sub-directories. These files are kept in top level of the Caucus home directory. These files include:

caucus_passwd	password file used by NCSA httpd
cmi_*	scripts used by the text-interface to integrate e-mail
credit	a full-screen visual editor supplied with the Caucus text interface
credit.doc	installation instructions for credit
csetperm	script to set (or correct) file permissions for most Caucus files
cv2	standard script to run the Caucus text interface
cv2cap	script to run Caucus text interface in "captive" mode
cv2check	script to check for new information in Caucus conferences
cv2kill	script to delete Caucus users
cv2mkmd	script to compile Caucus text-interface "dictionaries"
cv2pass	script to manage "captive" mode users

cv2remov	script to delete Caucus conferences
cv2start	script to create a new Caucus conference
expuser	script to delete "expired" users
fixdate	script to update date or "SINCE" information about old conferences
fixmaster	script to automatically corrected corrupted C**nnn**/masteres files
fixnames	script to rebuild corrupted MISC/names**nnn** files
fixtext	script to recreate missing TEXT001 sub-directories
listuser	script to list potentially "expired" users
master.opt	master options file for Caucus text interface
passprog	script to run cpw1 program to modify caucus_passwd or other httpd password file
manager_script	Caucus Management Menu
register	script to pre-register one or more users
swebstop	script to stop all running Caucus web-interface processes (swebd, etc.)
testconf	script to test consistency of conference item & response data
vvtermcap	file used by "credit" editor
webreg	script used to register new web interface userids and passwords

Chapter 7

Customizing the Web Interface

7.1 Introduction

This chapter describes some simple changes that may be made to the default Caucus interface. Although you may customize Caucus at any time, the changes described in this chapter would typically be made at the point of installation, as you are considering what Caucus defaults to set, how to customize Caucus to match the visual look of your Internet site or your intranet, and how to give access to Caucus to your users.

Caucus may be customized in two distinct ways. You can customize Caucus by changing system "switches," substituting your own icons, buttons and images for our defaults, inserting your own logo, and by editing the html pages which give access to Caucus. **Changes of this kind are discussed in the present chapter.**

You may also customize the Caucus interface in far more radical and thorough-going ways, including writing your own Web interface, to meet the needs of your organization. The Web interface to Caucus is written entirely in a simple interpreted programming language called CML, the Caucus Markup Language. CML is an extension of HTML, the Hypertext Markup Language, which includes programming constructs (conditionals, loops, etc.) and access to the Caucus database. CML scripts or "pages" generate dynamic "on-the-fly" HTML pages which then appear on a user's Web browser.

Screen Porch distributes a default set of CML pages that make up the standard Web Caucus interface. Each site is welcome to customize these pages, or even to create completely different interfaces. (A single site may have many different interfaces to the same set of conferences.)

For more information on CML, see section 6.4 of this guide and the separate CML Reference Guide. A CML Programmer's Guide is also forthcoming from Screen Porch. It will be available for download from our site. Keep in touch with "News" at the Screen Porch web site (http://screenporch.com) for more information. Customizing Caucus with CML is also under active discussion in the technical support conferences at our web site. Please make use of this important resource.

7.2 Caucus Web interface "switches"

Most sites will want to consider and perhaps change several Caucus options. These options are called "switches," because their values either turn a particular option "on" or "off," or else supply a particular value for a string (a file name, for example).

One example: you use a Caucus "switch" to specify the graphic image for the page background at your Caucus conference center.

There are two kinds of Caucus switches: those that control the configuration of the Caucus ("swebd") server program, and those that control options in the CML interface files.

7.2.1 Configuration Switches

The configuration switches are contained in the file **SWEB/swebd.conf**. This is an ordinary ("plain text") file which you can modify with an ordinary text editor. The file contains detailed comments that explain each of the options. **Note**: if you change this file, you **must** restart the Caucus server in order for your changes to take effect.

If you are upgrading your Caucus software, the installation procedure will create a file called **swebd.conf.new**. You should compare this file with your existing **swebd.conf** file, there may be new switches that you will wish to use in **swebd.conf**.

7.2.2 CML Interface Switches

The CML interface option switches are located in the file **CML/SP31/Local/switch.i**. Simply edit this file with a text editor to change the switches. Please see the detailed comments in **switch.i** for a list of the switches and how to set each one.

The default values for all of these switches is kept in the file **CML/SP31/defaults.i**. This enables you to return to the default values at any time by copying this file to **CML/SP31/Local/switch.i**. The values in switch.i override these defaults.

As you install new releases of Caucus, however, new switches may appear in the new **defaults.i** file. You can then copy these new switches to the **switch.i** file and set their values to meet your preferences.

7.3 Caucus Icons, Buttons, and Logos

7.3.1 The 'img' directory

Caucus includes a collection of icons, buttons, and logos that are stored as gif and jpeg files. These files are all stored in a single directory for convenience. The name of that directory is set

in the file **CML/SP31/startup.i**. Look in that file, and you will see a line that looks something like:

```
set img /~$caucus_id()/GIF31
```

This line defines a CML variable called "img" that contains the location (the directory containing) the gif and jpeg image files. In this case, the image files are stored under the Caucus home directory, under the sub-directory **public_html/GIF31**.

If you wish to change some or all of these image files, the best approach is to create a new directory and put the new image files in that directory. For example:

1. Create a new directory, such as **public_html/GIF4**.
2. Copy all of the files in **GIF31** to **GIF4**.
3. Put your changes in **GIF4**.
4. Edit **CML/SP31/startup.i** to point to **GIF4**, as in:

```
set img /~$caucus_id()/GIF4
```

It is **not** a good idea simply to change the images in **public_html/GIF31** directly. Your changes might then be wiped out the next time you install a Caucus upgrade.

7.3.2 Changing Individual Graphics

If you wish to just change a few graphics, it may be easier to create new files (with new names, and put them in **GIF31** with the other graphic files.

In that case, look in the file **CML/SP31/defaults.i**, and you will find the definition of a set of CML variables that reference the pre-defined logos and icons. Copy the definitions of the relevant graphics to **CML/SP31/Local/switch.i**, and then change the definitions there to point to your new files.

7.4 Caucus.html

There is one special HTML file that you may also wish to customize. It is located in the **public_html** directory.

Caucus.html is the template access-to-caucus HTML page. It provides links to the conferences, and also to the "self-register a userid" pages. You may wish to completely rewrite this page, or else extract the links and place them in the appropriate places in other HTML pages for your organization.

7.5 Changing HTML text in CML files

If you wish to make more extensive modifications to the interface, you can edit the HTML text inside the .cml files in **CML/SP31**. Before undertaking such edits, please copy all the files in this directory to a backup directory, in case you need to restore the originals.

All lines that produce HTML text must begin with a double-quote ("). Everything after the double-quote is HTML, or HTML mixed with CML functions (that eventually evaluate into plain HTML text).

7.6 Creating Distinct Interfaces

If you wish to create a very different interface, or wish to provide multiple interfaces (such as your own, and the default SP31 pages), you should create separate interface directories.

For example, to create a new interface called **XYZ** that is loosely based on the SP31 pages, follow these steps:

1. Create the directory **CML/XYZ**

2. Copy everything in **CML/SP31** (including the sub-directory **Local** and its contents) into **CML/XYZ**.

3. Edit the files in **CML/XYZ** to create your new interface.

4. Create a new .html file, based on public_html/caucus.html, to provide a URL that points to your new interface.

7.7 Upgrading Caucus

New releases of the Caucus software, and of the default **SP31** Caucus interface pages, will appear regularly on the Screen Porch home pages.

Currently, when you install a new release, the installation script renames your CML/**SP31** directory to CML/**SP31date**, where **date** is today's date.

If you have only made changes to the files under the "Local" subdirectory, or to the graphics, then you should copy everything from CML/**SP31date**/Local to CML/**SP31**/Local.

If you have made more extensive changes to the interface, you have three choices:

1. **Ignore the new release**. Rename CML/**SP31** to something else, and then rename CML/**SP31date** back to CML/**SP31**.

2. **Apply your changes to the new release**. Whatever changes you made before, make them again to the files in CML/**SP31**. This requires that you keep track of the changes that you made. (Or that you kept an unchanged copy of the original **SP31** files, and then you can use a utility such as diff or merge2 to find your changes.)

3. **Apply new features in the new release to your pages**. This is the inverse of choice number 2. Here you would use diff or merge2 to find the new features in the new **SP31** pages, and apply them to your pages.

All of these choices have disadvantages. This is a classic "fork" dilemma in software development, that happens whenever two or more parties add features to an existing product. The best advice is to ***always*** keep a detailed history of all changes that you make to your site, and to keep a copy of the unmodified **SP31** pages in a separate directory.

Appendix A

Pre-registering Caucus Users

A.1 Introduction

The first time that a person uses Caucus, they will normally be asked to "register" themselves. This means that they will be asked to enter their full name, and (optionally) a telephone number, a brief personal introduction, an e-mail address, and so forth. This registration process only happens once, although a user can change their personal information at any time.

Alternately, the Caucus manager for a site may choose to pre-register groups of users. This is particularly useful if you have a large body of users who are about to begin working with Caucus, and you want to enforce consistent use of names, phone numbers, e-mail addresses, and so forth for this new group. Pre-registration also allows you to automatically join groups of users to specific conferences, which may ease the users' confusion about which conferences are important to them.

A.2 Using the REGISTER script

To pre-register a batch of users, login to the Caucus userid, and run the **register** script.

To use **register**, you must specify a filename on the command line. For example, to register users from the file **prereg**, type "**register prereg**". The file, "**prereg**" must be in the home directory of the Caucus account and contain the pre-registration information described below.

A.3 File Format

The file which contains the list of users must be in a specific format. This format includes some necessary information and some optional information. The following information must be included:

Full Name
Last Name
User ID (either computer account ID or captive Caucus ID)
Telephone number

The following information may also be specified:

Default settings
Personal Introduction
User Variables
Initial conferences
Comments

The file may include information for as many users as the host computer system will support.

The information for each person must be supplied on 2 or more lines. These lines must be together in the file. The first of these supplies the 4 or 5 bulleted items in the mandatory list above, e.g.:

Jeff Victor:victor:jvictor:(313)-482-8710:

Note these rules about the first line of a person:

- Each field must be *terminated* by a single colon.
- No field can have a colon in it. This is especially important for the User Name and Telephone Number fields. The other fields shouldn't have a colon in them anyway.
- The only thing that distinguishes the fields is their order on the line!
- The first and fourth fields are free-text fields, and can contain Roman characters, kana, and/or Kanji. For example, the phone number can be anything. The second field must be one word, but can contain Romanji, kana, and/or Kanji. The third field must be one word, lowercase ASCII, with no spaces.
- The phone number can be blank, but the (empty) field must still be terminated by a colon.

So, the official technical specification for this line is:

Full Name:last_name:userid:phone:

The rest of the information lines for a person each begin (in the first column) with a special letter indicating the type of information supplied by that line. This letter must be uppercase.

The other mandatory line is just a line indicating that this is the end of this user's description. It is a single 'E' in the first column.

Here is an example of the simplest description:

Jeff Victor:victor:jvictor:(313)-482-8710:
E

A.4 Optional Information

In the order described above:

1) The Settings line must begin with an 'S'. This line must be in the same format as the "sys_Tset0" string in the dictionary file "america0". Any legal values can be specified. If the manager chooses to have the user start off with the 'normal' default settings, the line looks like this:

    Soff 23 79 off caucus 8 12 default on 0 . off wordwrap 0 later execmenu

2) A Personal Introduction can be many lines long. Each line must begin with an uppercase 'I'. Each line may contain Romanji, kana, and/or Kanji.

3) User Variables are one line each. Each line must begin with an uppercase 'U'. This must be followed immediately by a digit from zero to nine. The rest of the line can be followed by free-text. Each line may contain Romanji, kana, and/or Kanji.

4) Initial conferences are one line each. Each line must begin with an uppercase 'C'. This must be followed by the entire conference name. Although initial substrings of conference names can often be used, if an ambiguous name is used, the user will not be joined to the conference, the program will complain to the manager, and then it will continue.

5) You can insert comment lines anywhere in the file. To signify that a line is a comment and should be ignored by the maintenance programs, start it with a '#' ("pound sign", or "hash mark"). Each line may contain Romanji, kana, and/or Kanji.

A.5 Examples

A complete example:

```
#This is the first user:
Jeff Victor:victor:jvictor:(313) 482-8710:
Soff 23 79 off caucus 8 12 default on 0 . off wordwrap 0 later execmenu
IThis is a user's brief introduction.
IIt is two lines long.
U1This is the value of the User's Variable $(U1)
U8This is the value of the User's Variable $(U8)
# This is a comment
# The rest of the lines for this person are conferences to join the user into:
Cdemonstration
Csales
E
```

```
#
#This is the second user.  This is a captive user.
Charles Roth:roth:roth:(313) 482-8710:
#We'll accept the default settings.
IThis is the introduction.
Cdemonstration
E
```

A.6 Errors

If the program detects an unrecoverable error about a particular user (e.g. that ID already exists), it will complain about the user, not register him/her, and attempt to continue.

If the program thinks that the disk is full, it will mention this and then give up.

The software tries to protect the user from most mistakes, but it's possible to *really* confuse it, in which case it may complain and give up. On the other hand, if you include a colon as part of a user's name, it may fail in strange ways.