iChair Manual

Matthieu Finiasz and Thomas Baignères

– written around Christmas 2005 –

Contents

1 Introduction - What is iChair?

iChair is a powerful submission/review server software designed to help the program chair of a conference with:

- submission collection
- assignment of articles to reviewers
- review collection
- discussions
- mailing to authors and reviewers
- plenty of other things!

iChair was developed with a view to being as easy to install, configure and use as possible. Installation is explained (in details) in Section 2. Once installed, iChair is composed of three parts:

- the *administration* part, accessible only to the server administrator and used to configure iChair, manage the members of the program committee... Usage of this part is detailed in Section 3.

- the *submission* part, accessible to everyone and used to submit/revise/withdraw articles to the conference

- the *review* part, accessible to members of the program committee and to the observers. It is used by the reviewers to download submitted articles, to upload their reviews and to discuss which articles should be accepted. In this part program chairs have special privileges, giving them access to more features than other committee members: send mass mails to the authors/committee members, decide which articles should be accepted...

2 Installation

iChair was developed to run on a Linux box, using an apache 2 web server and PHP 5. The first official server (for Eurocrypt 2006) was running on a Gentoo Linux on a 1.8GHz Pentium 4 with 512MB of memory. At that time, our development server was running on an openSUSE 10.0 with a 1.4GHz Pentium 4 with 384MB of memory.

2.1 What you need before starting to install iChair

The installation of iChair requires a computer with Linux running (it might be possible to run iChair on a windows server or on macOS X, but this has not been tested yet) and a few packages installed. Here is a detailed list of what you need:

- `gs` – the ghostscript command.

- `pdf2ps` – a tool to convert pdf files to postscript. It normally comes bundled with `gs`.

- `psselect` – a tool to manipulate postscript. It usually comes inside a package named `psutils`.

- `libxml2`, `libxslt` and `libgcrypt` – some libraries required by PHP. As we intend to compile PHP from its sources you also need the sources of these libraries (for SUSE, this is the `-devel` versions of the package).

- `openssl` – the openSSL library. This library is only required if you intend to use SSL on your apache 2 server. We recommend installing it (see section 2.6 for more details) for more security. As for the previous libraries, as we intend to compile our apache web server from its sources, we will need the `-devel` version of this package.

- `sendmail` – a daemon used to send mail. Sendmail has to be installed and **running**. If sendmail is installed but not running, everything will install correctly, iChair will run normally, but no mail will be sent: the mail will remain in the sendmail queue, waiting for a running daemon to send them. You can check that the sendmail daemon is running with the command:

  ```
  ps -aef |grep sendmail
  ```

  You should normally obtain an output similar to this:

  ```
  root 1554 1 0 Oct13 ?  00:00:00 sendmail: accepting connections
  mail 1560 1 0 Oct13 ?  00:00:00 sendmail: running queue: /var/spool/mqueue
  ```

  We recommend adding `sendmail` to the default runlevel init scripts so that if the server has to be restarted you will not forget to restart the sendmail daemon.

- `zip` – the zip command. iChair can run without this, but it is required if you want reviewers to be able to download a zip of the articles they have to review.

- `pdftk` – the pdf toolkit. iChair can run without this, but it is required if you want to tag submitted PDF with the name of the conference (or any other watermark) making it easier for the reviewers to know for which conference they are doing a review. This program is not included in all distributions (it is not included in the SUSE for instance, but is in the Gentoo) and can be a little tricky to compile from the sources. . . You can visit the pdftk home page for more information.

2.2 Installing Apache 2

Apache 2 can often be installed directly with your Linux distribution. We do not recommend doing this as you usually have no control on the options with which apache is compiled. We recommend downloading the source tarball directly from the apache web site. Go to the download section, choose an appropriate mirror and download the UNIX source, in `.tar.gz` format. You should have a file with a name similar to `httpd-2.2.0.tar.gz`.

1. Go to the directory where you have downloaded `httpd-2.2.0.tar.gz`

   `cd download` (if you have downloaded the file to the `download` directory)

2. Uncompress and untar the apache archive

   ```
   tar xzf httpd-2.2.0.tar.gz
   ```

3. Enter the apache source directory

   ```
   cd httpd-2.2.0
   ```

4. Run the `configure` script which will customize the apache installation script to fit your Linux installation. The `--enable-so` option is required by PHP.

   ```
   ./configure --enable-so
   ```

   If you intend to use SSL for your server (see section 2.6 for more details) you should add the option `--enable-ssl`

   ```
   ./configure --enable-so --enable-ssl
   ```

5. Compile apache with the command

   ```
   make
   ```

6. You now have to log in as root to install the apache web server. You can run the commands

   `su` (you will be prompted for the root password, which you need to enter)
   ```
   make install
   ```

7. Now apache 2 is installed on your machine! If you have not changed the default installation path it should be located in the `/usr/local/apache2` directory. You can launch it with the command

   ```
   /usr/local/apache2/bin/apachectl start
   ```

8. If you want apache to start automatically every time your server is restarted you will have to add apachectl to the init scripts of the default runlevel of your server. You first need to add a link to the `apachectl` command in the `/etc/init.d` directory of your server. Still logged in as root, use the commands

   ```
   cd /etc/init.d
   ln -s /usr/local/apache2/bin/apachectl .
   ```

Once the link is created you need to tell your server to launch this service automatically at startup. The method to use will depend on the Linux distribution you chose. On an openSUSE we recommend using `yast2` to add `apachectl` to the default runlevel. Run `yast2`, select the `System` group, then the `System Services (Runlevel)` item. A list of services will appear and it should contain `apachectl`: enable it and click on finish. That's it.

On a Gentoo things are a little more complicated: the format of the `apachectl` script is not compatible with the startup script format of the Gentoo. You will need to write a king of "wrapper" to launch `apachectl`. You can use something like this and save it in the file `/etc/init.d/apache`:

```
#!/sbin/runscript
depend() {
    need net
}
start() {
    ebegin "Starting apache"
    /etc/init.d/apachectl start
    eend $?  "Failed to start apache"
}
stop() {
    ebegin "Stopping apache"
    /etc/init.d/apachectl stop
    eend $?  "Failed to stop apache"
}
reload() {
    ebegin "Restarting apache"
    /etc/init.d/apachectl restart
    eend $?  "Failed to restart apache"
}
```

You can then add this new script to the default runlevel with the command:

```
rc-update add apache default
```

For other Linux distribution we leave it to you to see how things work!

2.3 Installing PHP 5

Now that apache is installed, you will need to install PHP 5. Once again, it is recommended to compile it from its sources which you can get from the PHP web site. Click the downloads link at the top and download the Complete Source Code of the latest PHP 5 version (in `.tar.gz` format). You will then have the possibility to choose a mirror from which to download php 5. You should get a file with a name looking something like `php-5.1.1.tar.gz`. Now the installation process will look a lot like that of the apache 2 web server.

1. Go to the directory where you have downloaded `php-5.1.1.tar.gz`

   `cd download` (if you have downloaded the file to the `download` directory)

2. Uncompress and untar the PHP archive

   `tar xzf php-5.1.1.tar.gz`

3. Enter the PHP source directory

   `cd php-5.1.1`

4. Run the `configure` script which will customize the PHP installation script to fit your Linux installation. The `--with-apxs2` option tells PHP you are building a module for an apache 2 web server, the `--with-xsl` option is required to handle some XML features in iChair.

   `./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-xsl`

5. Compile PHP with the command

   `make`

6. You now have to log in as root to install the PHP 5 apache module. You can run the commands
   `su` (you will be prompted for the root password, which you need to enter)
   `make install`

Now PHP 5 is installed on your machine! You will simply have to configure apache a little to activate the module for `.php` files. This is explained in section 2.5.

2.4 Installing iChair

iChair needs to have access to some directories where it will save the articles submitted, the reviews of the program committee. . . The easiest way to do this is to create a dedicated user on your server: the iChair files and the submissions can then be saved in its home directory. The procedure for doing this is detailed hereafter.

1. The first step is to create a user `apache`, and choose its password. You need to be logged in as root to do this.
   `su` (type in the root password when prompted)
   `useradd -d /home/apache -m apache`
   `passwd apache` (type the password for user apache when prompted)

2. Now that the `apache` user is created, you can log in as `apache` and create some required directories: the directory where iChair will be uncompressed, the directories where the submissions and the reviews will be stored.

```
su apache (now you should be logged in as apache)
cd /home/apache
mkdir www (the directory from which iChair will be served)
mkdir MyConference2018 (the directory where all the files for your conference will be stored)
cd MyConference2018
mkdir submissions reviews logs
```

3. Now that all the necessary directories have been created, you simply need to uncompress and untar the iChair package. Suppose that `ichair.tar.gz` file is located in the `/mydirectory/download` directory.
   ```
   cd /home/apache/www
   tar xzf /mydirectory/download/ichair.tar.gz
   ```

That's it! Now if the apache web server is run by the user apache it should have the right to write all the files it needs, either to the `submissions`, `reviews` or `logs` directories, or to the `www` directory.

2.5 Configuring Apache 2

This part is probably the most difficult step of the iChair installation. However, as you will see, it is not that hard to configure a basic apache web server. Note that this description is valid for apache 2.2. If you are using an older apache 2.0 version things will be similar, but a little different!

1. All the configuration of apache is done in the file `/usr/local/apache2/conf/httpd.conf`. Only root can modify this file. Log in as root and use your favorite editor (say `emacs`) to edit the file
   ```
   su (type in the root password when prompted)
   emacs /usr/local/apache2/conf/httpd.conf
   ```

2. First you need to tell that `.php` files need to be treated as PHP. Add (at the beginning of the file for example) the two lines
   ```
   AddType application/x-httpd-php .php .phtml
   AddType application/x-httpd-php-source .phps
   ```

3. Then you need to tell the server to run under the username `apache`. Locate the two lines
   ```
   User nobody
   Group #-1
   ```
   and replace them by
   ```
   User apache
   Group users
   ```
   This way the server will have the right to write files in the `/home/apache` folder.

4. Locate the line
   ```
   ServerAdmin you@example.com
   ```
   and replace the email address by the address you want to appear in the error pages generated by apache.

5. You now need to change a few apache options for the directory `/home/apache/www`. This is necessary for iChair to be able to configure the apache web server behavior through `.htaccess` files. Locate the line
   ```
   <Directory "/usr/local/apache2/htdocs">
   ```
   and just before this line insert the following lines

```
<Directory "/home/apache/www">
    Options None
    AllowOverride Options AuthConfig Limit
    Order allow,deny
    Allow from all
</Directory>
```

6. Now, you need to tell apache to also serve `index.php` files when requesting a directory index. Locate the line

```
DirectoryIndex index.html index.html.var
```

and complete it to obtain

```
DirectoryIndex index.html index.html.var index.php
```

7. The final step is to tell apache to serve the files in the directory `/home/apache/www` when someone connects to your server. If you intend to use SSL for your server you should not perform this last operation, otherwise it will also be possible to access iChair without using SSL. In that case you can directly jump to the next section on using SSL (section 2.6). If you do not want to use SSL, just locate the line

```
DocumentRoot "/usr/local/apache2/htdocs"
```

and replace it with

```
DocumentRoot "/home/apache/www"
```

8. Save the file `/usr/local/apache2/conf/httpd.conf` and exit your favorite editor.

If you do not want to use SSL, the configuration of apache is finished. You now have a fully functional web server, ready for a use with iChair! You can now jump to the section 3.

Do not forget to restart apache as the modifications will not be taken into account otherwise. This can be done with the command

```
/usr/local/apache2/bin/apachectl restart
```

2.6 If you want to use SSL

We strongly recommend using SSL for your iChair server: the members of the program committee will have to login with a user name and password to access their reviews and unless you use SSL, these will be **transmitted in clear**.

Enabling SSL on your apache server is done in three steps:

1. Generate (or obtain from a certified authority) a certificate for your server,

2. Configure the `httpd-ssl.conf` file to serve iChair files,

3. Restart the apache server with SSL enabled.

2.6.1 About certificates

If you want to run an SSL server, you need a certificate on this server, this certificate will be sent to the users connecting, giving them the public key of the server and letting them encrypt their request so that only the server (the only one knowing the associated secret key) can decipher it. Their are basically two ways to get a certificate: either you obtain it from a valid authority (thawte, verisign...), or you generate a self signed certificate for your server. The first solution will cost you money and can take some time, with the second solution the users will be warned by their browser that the certificate could not be verified.

Here we will describe how to use the second solution and generate a self signed certificate using openssl.

1. As usual, you will have to be logged in as root to install the certificate.

    `su` (type in the root password when prompted)

2. The certificates will be placed in the `conf` directory of your apache web server. You will first need to generate the secret/public key pair for your server. We believe an RSA 2048 key is well suited

    ```
    cd /usr/local/apache2/conf/
    openssl genrsa 2048 > server.key
    chmod 400 server.key
    ```

3. Now that you have a key, you can generate a self-signed certificate

    ```
    openssl req -new -x509 -nodes -sha1 -days 365
                                        -key server.key > server.crt
    ```

   - `req` means you are requesting a certificate
   - `-new` means you want a new certificate
   - `-x509` tells `openssl` to generate a self signed certificate instead of a certificate request (which you would have to send to a third party authority for it to be signed)
   - `-nodes` disables private key encryption. If you do not use this option, you will have to enter a password every time you launch apache with SSL (including at each server reboot). Encrypting the password is only useful if some untrusted persons might have access to the server
   - `-sha1` tells `openssl` to use SHA-1 for signatures (other choices are MD5, MD2 and MDC2).
   - `-days 365` determines the number of days until certificate expiration. Be sure to specify a number large enough to reach the end of the review process, otherwise you will have to regenerate a certificate at some point.
   - `-key (...)` specifies the secret/public key pair to use for the certificate. Here we use the key we generated in the previous step.

    You will then be prompted for the values of various fields of the certificate

   - `Country Name (2 letter code)` – enter the two letter country code for the country in which the server is located. This should follow ISO 3166-1 specification. For Eurocrypt 2006 this was "CH"
   - `State or Province Name (full name)` – the full name of the state/province in which the server is located. For Eurocrypt 2006 this was "Vaud"
   - `Locality Name` – The city in which the server is located. For Eurocrypt 2006 this was "Lausanne"
   - `Organization Name (eg, company)` – the name of the company/university in which the server is located. For Eurocrypt 2006 this was "EPFL"
   - `Organizational Unit Name` – this could be the name of the research team hosting the server. For Eurocrypt 2006 this was "LASEC"
   - `Common Name (eg, YOUR name)` – contrary to what is indicated, you should NOT enter your name here, but the name of the server: `server-name.domain-name.ext`. For Eurocrypt 2006 this was "lasecpc11.epfl.ch"
   - `Email Address` – the email address which will be included in the certificate.

    That's it, the certificate has been generated successfully, you will now have to configure apache to use this certificate. This is done in the `httpd-ssl.conf` file.

2.6.2 Configuring `httpd-ssl.conf`

1. As for the configuration of apache with the `httpd.conf` file, you will have to be logged in as root and use your favorite text editor to adit the `httpd-ssl.conf` file.

   `su` (type in the root password when prompted)
   `emacs /usr/local/apache2/conf/extra/httpd-ssl.conf`

2. In this file we will configure a *virtual host*, listening on port 443 of the server (the default SSL port) and using SSL encryption. Locate the line

   `<VirtualHost _default_:443>`

   This line marks the beginning of the description of the virtual host. Just below, you should find three lines looking like

   `DocumentRoot "/usr/local/apache2/htdocs"`
   `ServerName www.example.com:443`
   `ServerAdmin you@example.com`

   Modify the `DocumentRoot` line to

   `DocumentRoot "/home/apache/www"`

   and modify the two other lines to match the name of the server and the email address which will appear in the error pages generated by apache.

3. A little further in the file should appear a line looking like

   `SSLCertificateFile /usr/local/apache2/conf/server.crt`

   Verify that the file name specified in this line is exactly this one so that it points to the certificate file we generated previously.

4. Then the line

   `SSLCertificateKeyFile /usr/local/apache2/conf/server.key`

   Once again, verify that the file name is correct

5. Save the `httpd-ssl.conf` file and exit your text editor.

That's it, SSL is configured on your server. In previous versions of apache (up to version 2.0), starting apache with SSL enabled was done with the command

`/usr/local/apache2/bin/apachectl startssl`

However, with apache 2.2 this option is no longer supported and strangely enough, this makes things much easier!

2.6.3 Enabling SSL in apache

To enable SSL you simply need to modify the `httpd.conf` file to tell it to include the `httpd-ssl.conf` file we just edited.

1. As always you will have to be logged in as root and use your favorite text editor to edit the `httpd.conf` file.

   `su` (type in the root password when prompted)
   `emacs /usr/local/apache2/conf/httpd.conf`

2. Now simply locate the line (it should be the 10th line from the bottom of the file)

   ```
   #Include conf/extra/httpd-ssl.conf
   ```

   and simply uncomment it by removing the # character to obtain

   ```
   Include conf/extra/httpd-ssl.conf
   ```

3. That's it, save the `httpd.conf` file.

Now every time apache is restarted it will also start the SSL server on the port 443. Try to stop and then start apache and verify that you can connect to the server in `https`.

```
/usr/local/apache2/bin/apachectl stop
/usr/local/apache2/bin/apachectl start
```

Use your favorite web browser to access the page `https://server-name.domain-name.ext/` (replace `server-name` and `domain-name.ext` by the name and the domain of the server). You will probably get a "Forbidden" error message but this is normal and it still means that your configuration was successful.

3   Configuring iChair

All the configuration of iChair is done online, using the *administration* part of the software. From now on we will consider that the server on which iChair is installed is accessible at the address `https://server-name.domain-name.ext/` (replace `https` by `http` if you did not use SSL and replace `server-name` and `domain-name.ext` by the name and the domain of the server).

3.1   Set an `admin` password

Use your favorite web browser to access the page
   `https://server-name.domain-name.ext/iChair/admin`

At the top of the page you should get a warning telling you that no *Admin Password* has been set yet. Follow the link in this warning to change it. As there is no password yet, you do not have to enter anything in the `Old Administrator Password` field. Just enter the same password in both `New Administrator Password` fields and click `Update Password`. When you click `Ok` on the next page, you will be prompted for a login and password. The login is always `admin` and the password is what you just chose.

3.2   Use the `configuration` page

Now that a password is set, you can start configuring iChair. The menu on the right of the page contains all the pages you need to configure. You should first visit the `Configuration` page.

Below is a description of all the fields you will have to fill to configure iChair. Do not forget to click the `Update Configuration File` button at the button of the page or your changes will not be saved.

- Main Conference Settings

  - `Conference Name`: the name of the conference, which will appear at the top of every page of iChair and in all the mails sent to the authors or the committee members. Try to keep it as short as possible ("ISPEC 2005" is a good name but not "The First Information Security Practice and Experience Conference").

  - `Conference Web Site`: the web page where information on the conference can be found (program, venue...).

– **Conference Logos**: for each logo enter the locations of the image to use, either an absolute path (starting with `http://`) or a relative path (starting from the `iChair/` directory), and the address to which one should be redirected when clicking one of this logo. See question 5 of the FAQ - Troubleshooting section for details about using more than 3 logos.

- **Server Settings**

  – **This Server Location**: enter the exact location of the iChair server which is running. You can simply cut'n'paste the address appearing in your browser and remove the end so that it finishes with `/iChair`.

  – **Submissions Path**: enter the directory in which submissions should be saved. It should be something like `/home/apache/MyConference2018/submissions/`. Remember that the user running Apache (this should be user `apache`) must have sufficient permissions to write in this directory.

  – **Reviews Path**: the directory in which the reviews (and last version of each submission) will be saved. It should be something like `/home/apache/MyConference2018/reviews/`.

  – **Log Path**: the directory in which the logs of iChair will be saved. It should be something like `/home/apache/MyConference2018/logs/`. Advanced administrators might want to redirect these log into the `/var/log/` folder but if you decide to do so, remember that the user `apache` must have the read/write access to the directory you chose. As iChair logs are not very large (about 1MB at the end of Eurocrypt 2006) this is not really necessary anyways.

  – **Path to the zip command**: if you want to enable on the fly zips (which will allow members of the program committee to download a zip of assigned articles) you should enter the directory in which the `zip` command you want to use is installed. Leave this field blank to disable on the fly zips. If submitted articles are very large, enabling on the fly zips can slow down the server. For Eurocrypt 2006, with articles of 500kB or so and 30 articles assigned to each reviewer, these on the fly zips did not cause any problem.

  – **Path to the pdftk command**: if you want to be able to tag submitted articles before members of the Programm Committee can download them, you will need to install `pdftk` (this might not be as easy at it seems...). Once `pdftk` is installed, you can then simply enter the path to the executable (like for the `zip` command) and fill in the field in the Review Setting part of the configuration to activate the tagging.

  – **Official Deadline Date** and **Official Deadline Time**: the official date and time at which the submission server should close (as written in the call for papers). These dates and time should be given in UTC time, which means you need to convert the call for papers deadline time if it is not given in UTC time. These date and time will be displayed on the submission server and used to compute the time left counter.

  – **Effective Shutdown Date** and **Effective Shutdown Time**: the effective shutdown date and time after which the submission server will refuse new submissions/revisions/withdrawals. If you decide to give a few bonus minutes, authors arriving on the submission site will see a "Time is Up" warning in the time left box, but will still be able to submit/revise/withdraw articles until the effective shutdown.

  – **Server Shutdown Message**: the message which should be displayed instead of the submission/revision/withdrawal pages, after the effective shutdown time.

  – **Max Submission File Size**: the maximum size for a submission or a review or any other file uploaded to the server. Leave blank if you do not want to override de

default values specified in the `php.ini` file on the server. In PHP, there are three different variables which can modify this maximum file size: `upload_max_filesize` which is exactly the maximum size of an uploaded file, `post_max_size` which is the maximum size for a whole POST operation (in iChair we use `upload_max_filesize` + 3M) and finally `memory_limit` which is the maximum memory a PHP script can use. This last option will only be taken into account if PHP was configured with the `--enable-memory-limit` option but if this was the case, you will need to edit the `php.ini` file to modify its default value.

- **Time Display Format**: choose whether you prefer times to be displayed in 12h or 24h format.

- Mail Settings

  - **Admin e-mail**: enter the e-mail address of the administrator of iChair. This e-mail is displayed at the bottom of the submission part pages. If you want to use multiple addresses you can simply separate them with "," (comma).

  - **Global BCC e-mail**: check this box and fill in an e-mail address of your choice if you want *every* mail sent by iChair to be sent in BCC to a specific address. This can be very usefull to keep an archive of all e-mails sent, especially in case some contact authors have problems with their e-mail. You might for example want to save all e-mails locally on the machine by sending them to an address like `apache@server-name.domain-name.ext`. If you want to use multiple addresses you can simply separate them with "," (comma).

  - **"From" address used in mails to authors**: enter the e-mail address from which automatic submission/revision/withdrawal mails should appear to come from. If a contact author replies to one of these mails, he will reply to this address. This address is also used for mails sent by the program chair to the authors (notification, reviews). This should be the address of a person ready to receive quite a few emails from unhappy authors, so maybe the address of the program chair. If you want to use multiple addresses you can simply separate them with "," (comma).

  - **"From" address used in mails to reviewers**: the e-mail address from which mails sent to members of the Program Committee should appear to come from. This should probably be the e-mail address of the program chair. Again, if you want to use multiple addresses you can simply separate them with "," (comma).

  - **Submission Subject** and **Submission Message**: enter the subject and main body of the mail which will be sent to contact authors when they submit a new article. You have the possibility to use "tags" (between [ ]) in these fields which will be replaced by configuration/user dependent information before being sent. The **Preview Mail** button lets you see what the mail would look like once the tags have been replaced and gives you a list of all accepted tags in these two fields. If you modify one of these fields, you need to save the configuration (the button at the very bottom of the page) before previewing the new version.

  - **Revision Subject** and **Revision Message**: same as submission, but used when a revision is sent.

  - **Withdrawal Subject** and **Withdrawal Message**: same as submission, but used when an article is withdrawn.

  - **Password Subject** and **Password Message**: same as submission, but for mails containing the login and password of reviewers. Once they get this mail, reviewers are able to access the iChair review section.

- Submission Settings

  – `Available Fields`: check the boxes corresponding to the fields that you want the authors to fill-in when submitting an article. The `Title`, `Authors` and `Contact e-mail` fields can not be deactivated. Only the `Affiliations`, `Abstract`, `Category` and `Keywords` can be removed. Note that if a field is checked, the author will not have the possiblity to leave it blank.

  – `Previews`: check the box if you want a jpg preview of the first page of each submission to be rendered. This makes it possible for authors to verify that the file they submitted is the correct one but might increase the CPU load of the server. The `Test PS Preview` and `Test PDF Preview` buttons let you check that previews work correctly on the server. This should be the case if `gs` and `psselect` have been correctly installed. If you use version 7 of ghostscript, the PDF preview will probably be rotated, but this will only affect articles written in landscape format, which is not often the case!

  – `Categories`: if you chose to require a category for the submissions, this is where you can type your category list. These categories need to be entered in XML format. You have the possibility to create category groups and stand alone categories which are not part of any category group. Each category group has a name and contains several categories. The example below should be enough to understand how this works.

```
<categoryGroup>
  <name>martial arts</name>
  <category>karate</category>
  <category>judo</category>
  <category>kung-fu</category>
</categoryGroup>
<categoryGroup>
  <name>ball sports</name>
  <category>football</name>
  <cateogry>baseball</name>
  <category>handball</name>
</categoryGroup>
<standAloneCategory>ski</standAloneCategory>
<standAloneCategory>diving</standAloneCategory>
```

- Review Settings

  – `Print the specified watermark on all articles`: check the box if you want to tag all the submissions before members of the Programm Committee can download them. You can enter the tag/watermark you want to use in the textfield. Please note that article tags (the same tags as in the mails, still written between []) can be used. A tag like `[conf] - Submission [article_number] - Do not distribute` will print the name of the conference and the submission number on each article. This option should always work on `.ps` files but will require to have `pdftk` installed on your system to work on `.pdf` files. Note that this option is not very robust and the tagging might not work on some submissions.

  – `During Election Phase authors have access to the zip of all articles`: check this box if you want reviewers to be able to download all the articles when choosing which articles they want to review. This might help them with their choice, but will also probably increase the time it takes for them to make their choice...

  – `Default Number of Reviewers per Article`: enter the default number of reviews you want for each article. This will be the number of reviewers which will be assigned

to review each article. In average, each committee member will have to review:

$$\frac{\text{Number of Articles} \times \text{Number of Reviewers per Article}}{\text{Number of Reviewers}} \quad \text{articles.}$$

You have until the end of the submission phase to set this number. Once you have loaded submissions for review, you will need to modify the number of reviewers individually for each article.

– `Default Number of Reviewers per Article from a Member of the Program Committee`: the same as above, but for articles which were submitted by a member of the program committee. The program chair have the possibility to mark articles as coming from a committee member during the submission phase.

– `Hide Author Names to the Reviewers`: uncheck this box if you want all the reviewers to have access to the names of the authors. If this box is checked, only the program chair will have the possiblity to see the authors.

– `Hide Reviewer Names to the Authors`: uncheck this box if you want the authors to know who wrote which report on their article. If this box is checked, the authors will only get the contents of the reports, but won't know who wrote them (which might be a safer solution).

– `Preferred Articles Deadline Date` and `Time`: this time is only an indication for the reviewers, it will let them see how much time is left for them to finish choosing which articles they want to review. This should correspond to the date and time at which the program chair intends to perform an automatic assignation of the articles.

– `Review Deadline Date` and `Time`: this time is only an indication for the reviewers, it will let them see how much time is left for them to complete their reviews. After this date most reviews should be finished and a discussion phase can start.

– `Review Fields`: use the checkboxes to choose which grades and fields should appear in the reviews. The overall grade cannot be removed as it is used to compute averages and sort the articles. For each grade you have the possibility to select a range of valid grades (integer values from `Min` to `Max`, negative grades are not allowed) and supply a semantic of what each possible grade means. For example, if you set overall grade to go from 0 to 4 and consider 0 means reject, 4 means accept and 2 means you have no opinion, the semantics should be written:

```
0 - reject; 2 - no opinion; 4 - accept
```

• Miscellaneous

– `Notification Date`: the date at which you intend to notify authors whether their article was accepted of rejected. This date is only used to replace the `[notification]` tag which can be used in some automatic mails.

– `Final Version Date`: the date at which the final version of the paper is due. This date is only used to replace the `[final_version]` tag which can be used in some automatic mails.

– `New visit timeout`: the reviewers have a possibility to have their last visit date set automatically by iChair everytime they leave the iChair server page. If a user does not load any new page for the time determined by this timeout, then his last visit date will be automatically updated.

4   Understanding what each page does

5   Frequently Asked Questions - Troubleshooting

$\boxed{Q_1}$ **Why do all the pages in iChair look ugly?**
There are two possibilities: either you do not like the design we chose for iChair or the browser you use does not understand CSS correctly. iChair has been tested and displays correctly on the following browser:

- firefox 1.0
- safari (macOS X)
- konqueror 3.2
- opera 8
- mozilla 1.7 (version 1.6 makes some minor display errors)
- internet explorer 6 (however CSS popUps are not displayed)
- amaya (it is ugly, but everything seems to work)

It does not work on the following browsers:

- lynx (unable to post submissions)
- dillo, links (no support for HTTP authentication)
- netscape 4.79 (bad CSS interpretation making iChair unusable)
- emacs-w3 (submissions work, but no CSS support)

$\boxed{Q_2}$ **What login should I use to access the administration pages?**
Unless you changed it as explained in question 3 the default administrator login is `admin`.

$\boxed{Q_3}$ **How can I change the administrator login from `admin` to something else?**
This cannot be done through the iChair web interface. You need to log on the iChair server as user `apache` and edit the `.htpasswd` file in the `admin` directory.

```
su apache
cd /home/apache/www/iChair/admin
emacs .htpasswd
```

The file contains only one line looking like

```
admin:61fjTWIs8yZEU
```

Just replace the login `admin` by whatever you want (be sure not to add any space, remove the : or modify the password hash in the second part of the line) and save the file. Next time you will try to access an administration page you will have to enter the new login, using the same password as before.

$\boxed{Q_4}$ **I am administrator and I forgot the admin password, what should I do?**
Nothing to worry about, the password can be reset by deleting the `.htaccess` and `.htpasswd` file in the `admin` directory. Log on the iChair server as user `apache`

```
su apache
cd /home/apache/www/iChair/admin
rm .htaccess .htpasswd
```

Now the admin password has been reset (in case you had changed the administrator login as explained in question 3, the login has also been reset to the default `admin`). Do not forget to visit the *Admin Password* page to set a new password!

|Q₅| **How can I display more than three logos in the submission pages of iChair?**
By default iChair only proposes up to three logos because we believe that adding more logos could be too much. However, if you really want to display four or more logos this is fairly easy. All the configuration of iChair is stored in an XML file on the iChair server. You simply need to edit this file and add new logo lines. After this you can use the normal configuration method (see section 3.2 for more details) to change these logos. Log on the iChair server as user `apache`

```
su apache
cd /home/apache/www/iChair/utils
cp config.xml config.backup (to make a backup copy of the config file)
emacs config.xml
```

Near the beginning of the file, locate the </logos> closing tag. Just before it insert the lines

```
<logo>
  <number>4</number>
  <img></img>
  <link></link>
</logo>
```

This will add the possibility for a fourth logo. Add more if you need more and save the file. When you go back to the `configuration` page you should have the possibility to choose a fourth logo. If you get error messages when accessing the page, you most probably inserted invalid XML in the `config.xml` file. Restore the backup copy of the file and try again.

```
su apache
cd /home/apache/www/iChair/utils
cp config.backup config.xml (to restore the backup)
```