Setting Up Zambia from Scratch

Percy, Bendyogagirl[*]

July 2011

Abstract:

[*]NELA.Percy@gmail.com

Contents

List of Tables

List of Figures

1 Before beginning

There are some required programs to run the Zambia software. The following programs must be running on whatever server is going to serve up your Zambia instance: (LAMP or WAMP (WAMP is untested))

- apache
- php
- mysql
- some form of email sending software (MTA)
- SFTP/SCP

Either on your staging machine (where you will be uploading the information from) or on your server (if you are loading directly to there) you must also have:

- svn
- text editing program

1.1 Definitions

Table 1: Acronyms and Definitions

Acronym or Term	Definition
precis	An element in your schedule. Could be a class, panel, gathering, party, room-coverage, lounge, or whatever else you might have on your schedule
LAMP	Linux + Apache + MySQL + PHP/Perl/Python
WAMP	Windows + Apache + MySQL + PHP/Perl/Python

1.2 Decisions

You will need to decide the following before you install initially:

1.2.1 Standalone/Combination

Is this system going to be a standalone system, or are you going to interface it with Congo, or another piece of registration software?

1.2.2 Single/Multiple installation

Is this going to be a single instantce of Zambia, or are you going to have multiple instances of Zambia running on the same machine with some shared resources?

1.2.3 Webserver

Where is your web installation location (whatever directory you will be putting your files into)? Often something like /var/www/Zambia-Con or /home/username/public_html/Zambia-Con or the like.

1.2.4 db_name.php file

This information will go in the db_name.php file when you install it.

- Required information for the program to run:
  - Database hostname (**DBHOSTNAME**). If you are setting things up on the same machine that MySQL is running on, *localhost* should be what you are using for **DBHOSTNAME**.
  - Database username (**DBUSERNAME**). The user created for the Zambia program to access the database(s).
  - Database password (**DBPASSOWRD**). The password created for the Zambia program to access the database(s).
  - Main database name (**DBDB**). The database for all the single convention specific information to live in.
  - Biographical information database name (**BIODB**). The database for all the (possibly shared between Zambia instances) biographical data. If you are having the same people either for multiple years, or for multiple activities, it makes sense to centralize the information, otherwise it could be a one-shot for the particular convention.
  - The Reports databasse name (**REPORTDB**) points to where the reports are kept. It is on of the main ways of interation with the data about the convention. If you have multiple Zambia instances it makes sense for this to be a single, shared resource.
  - The CongoDump database name (**CONGODB**). Congo is another piece of software designed to interact with Zambia. The information in that set of databases can be accessed in a variety of ways.
  - The Limits (**LIMITDB**) and Localizations (**LOCALDB**) databases are used to key things that are specific to a convention, but might all want to live in a central database, for ease of deploying new conventions.
  - The TimeCard database (**TIMECARDDB**) might want to be specific to a convention, or generalized across many different activities of your organization, it often is local to the convention instance, but there are some reasons to have it in a single, externally served database.
  - The convention database keying tag (**CON_KEY**) is useful when the Limits and Localizations are stored in a single table, to distinguish between which instance needs to be pulled.
  - your host name (**MYHOST**) (If you are setting things up on the same machine that MySQL is running on, *localhost* should be what you are using for **MYHOST**) (this doesn't actually go in the file)
- Extremely useful information: Most of this information, currently in the db_name.php file, would all be candidtates to be inthe Localization database.
  - Con name.
  - Zambia administrator email.
  - Brainstorm email.
  - Programming email.
  - Registration email.
  - Number of days the con will run (code works for 1-8 currently.)
  - Date and time the con will start (In the format of yyyy-mm-dd HH:MM:DD for parsing purposes. Suggested 00:00:00 for the start time.)

  - URL of the con (without the leading `http://` in the URL.)
  - Logo for the con (gif, png, etc.)
  - Availabilty Records (starting number of "availability" fields to render in the "when I am available" form, 8 is a good default.)
  - Are kids avaiable (This should probably be set to "FALSE", it is a hold-over from backwards compatibility.)
  - Default Duration of the classes. (How long, in H:MM format that the classes are expected to be.)
  - Duration in Minutes (Should probably be "FALSE": TRUE: in mmm; False: in hh:mm - affects session edit/create page only, not reports.)
  - Grid Spacer (The time divisions in the fixed grid produced, in seconds. For example 1800 is 60 sec/min and 30 min, and a good default.)
- Very useful information
  - Guests of Honor badgelist (if you have specific featured Guests of Honor, the badgeids get listed here, comma seperated.) (This is being migrated as a flag for a presenter.)
  - Prefered total number of sessions upper limit (so your presenters don't oversubscribe themselves 5 is good default for a 3-4 day con.) (This should move to the Limits database.)
  - Prefered daily number of sessions upper limit (3 is a good default.) (This should move to the Limits database.)
- Description minimums and maximums All of this information should be migrated to the Limits database. Only set a value if you need it, unset values are simply presume there is no limit to the information in that direction. At some point, the precis descriptions will be folded into the same, or a similar structure to the Biographical Information is, currently.
  - Minimum web biography character length (Some cons have minimum biographical information requirements.)
  - Maximum web biography character length (Some cons have a different limits for what is on the web, and what is in the book, 3000 characters is a good starting default for the WWW.)
  - Minimum book biography character length (Some cons have minimum biographical information requirements.)
  - Maximum book biography character length (if there isn't a difference in the limit, but still, there is a limit, set to the same as the web maximum.)
  - Minimum uri biography character length (the URI block is often unlimited in either direction.)
  - Maximum uri biography character length (the URI block is often unlimited in either direction.)
  - Minimum picture biography character length (the picture line is often unlimited in either direction.)
  - Maximum picture biography character length (the picture line is often unlimited in either direction.)
  - Minimum web precis character length (You need a precis description of at least this long to be acceptable, 10 as a good default.)
  - Maximum web precis character length (3000 as a good default.)
  - Minimum book precis character length (if there isn't a difference in the limit, set it to the same as above for these, as well.)

  - Maximum book precis character length
  - Minimum precis title character length (You need a precis title of at least this long to be acceptable, 5 as a good default.)
  - Maximum precis title character length (50 as a good default.)
  - Minimum name character length (if there is a need to make sure all names are of at least a specifc length.)
  - Maximum name character length (if there is a need to make sure all names are no more than a specifc length.)
- Other interesting settings The linguistic settings (below) will go away once the precis descriptions are migrated to a similar format as the Biographical information currently resides.
  - Is this a bilingual event (This should probably be set to "FALSE" due to it's lack of complete support across the system, and the next few elements in this list, ignored.)
  - What the second language is.
  - Title caption in second language.
  - Description caption in second language.
  - Biography caption in second language.
- The rest of the file should not have to change.

2 Downloading

If you are checking the code out directly on the hosting machine, replace the final "Zambia" with what you decided the web install location will be. If not, you can rename it to whatever you wish to call your staging area.

Please, check out the code from:

svn co `https://zambia.svn.sourceforge.net/svnroot/zambia/branches/FFF/` Zambia

3 Local file setup

There are certain localisms you want to set up, outside the svn tree. This is so (should you need to) if an upgrade to the code-base is desired, it can be done, without writing over your customizations.

To begin the process change to your web install location. There you should see a list of files, and you will add one called "Local". When done your list of files should be: These files you will have to create or modify.

3.1 db_name.php

Copy the example webpages/db_name_sample.php to Local/db_name.php as a start, and then put in the values you decided upon before starting the install process.

3.2 FooterTemplate.html

This is where you will put your standard footer, that will be below all the public pages, to customize the look and feel to match your event's presentation.

3.3 HeaderTemplate.html

This is where you will put your standard header, that will be above all the public pages, to customize the look and feel to match your event’s presentation.

3.4 Participant_Images

If you choose to have images of your participants with their bios, make this directory, and any pictures that match the badgename of the participant will be put next to the bios.

3.5 Verbiage

If you wish to customize what is put forth to your participants, many of the pages allow for customization. The list of them will grow as more are done. Any information in these files will replace the default text. Some examples of these files are:

3.5.1 BrainstormWelcome_0

```
<P> Here you can submit new suggestions or look at existing ideas for
panels, Meet and Greets, Special Interest Groups, Birds Of a Feathers,
Author Readings, and Author Signings.</P>
<P> As suggestions come in and we read through them, we will rework
them, combine similar ideas into a single item, split large ones into
pieces that will fit in their alloted time, etc.  Please expect the
suggestions you submit to evolve over time.</P>
<P> Also, please note that we always have more suggestions than are
physically possible with the space and time we have, so not everything
will make it.  We do save good ideas for future conventions.</P>
<UL>
  <LI> <A HREF="BrainstormSearchSession.php">Search</A> for similar
  ideas or get inspiration.
  <LI> Email <A HREF="mailto:program@ourcon.org">
  program@ourcon.org</A> to suggest modifications on existing
  suggestion.
  <LI> <A HREF="BrainstormCreateSession.php">Enter a panel, MnG, SIG,
  BOF, et al suggestion.</A>
  <LI> <A HREF="BrainstormSuggestPresenter.php">Enter a suggestion for
  a Presenter.</A>
  <LI> See the list of <A HREF="BrainstormReportAll.php">All</A>
  suggestions (we’ve seen some and not see others).
  <LI> See the list of <A HREF="BrainstormReportUnseen.php">New</A>
  suggestions that have been entered recently (may not be fit for
  young eyes, we haven’t see these yet).
  <LI> See the list of <A HREF="BrainstormReportReviewed.php">
  Reviewed</A> suggestions we are currently working through.
  <LI> See the list of <A HREF="BrainstormReportLikely.php">Likely to
  Occur</A> suggestions we are or will allow participants to sign up
  for.
  <LI> See the list of <A HREF="BrainstormReportScheduled.php">
  Scheduled</A> suggestions.  These are very likely to happen at con.
  <LI> Email <A HREF="mailto:vols@ourcon.org">vols@ourcon.org</A> to
  volunteer to help process these ideas.
</UL>
```

3.5.2 Introduction_Blurb_0

```
and before I introduce our speaker, let me ask, How many of you are
new?  Well, let me tell you, you are in for one heck of a ride.<br
```

3.5.3 Schedule_Blurb_0

```
Welcome to the Circus Fantastique.  We really appreciate all your
efforts to make this weekend go so well.  Below is your schedule for
the weekend.  If you have any questions, please, do not hesitate to
find our staff in the Green Room, or whomever our point-person is at
that time.  </P><P>I hope you will have all the fun you can!<hr>
```

3.5.4 StaffPage_0

```
<P> Please note the tabs above.  One of them will take you to your
participant view.  Another will allow you to manage Sessions.  Note
that Sessions is the generic term we are using for all Events,
Classes, Panels, BOF/SIG/MnG, other activities, etc. </P>

<P>Current roles:
<UL>
  <LI>Pre-con Logistics: That tall guy, with the ’stash
  <LI>At-con Logistics: Bill(1)
  <LI>Speaker Liaison: Kat (with a "K")
  <LI>Assistant Speaker Liaison: Bill(2)
  <LI>Volunteer Captain: Cat
  <LI>Assistant Volunteer Captain: The Other Cat
  <LI>Green Room Czar: Tim
  <LI>Point People: Helium 1, Helium 2, and the Stupid But Cute.
  <LI>Schedule Wranglers: The group as a whole
  <LI>Technical support: Will
  <LI>(Tentative) Technical Support: Nyot
  <LI>Bio/copy editing: Rupert
</UL></P>
```

```
<P>The general flow of sessions over time is: <UL> <LI>Brainstorm -
New session idea put in to the system by one of our brainstorm
users. The idea may or may not be sane or good.  It could be too big
or too small or duplicative.

  <LI>Edit Me - New session idea that a participant or staff member
entered.  An idea entered by a brainstorm user that is non-offensive
should be moved to this status.  These are still rough and may well
have issues.  Still could be duplicates.

  <LI>Vetted - A real session that we would like to see happen.  At
this point the language should be fairly close to final in the
description. Spell checking and grammar checking should have happened.
It needs have publication status, a type, kid category, division and a
room set.  Please check the duration (defaults to 1 hour) and the
various things the session might need (like power, mirrors, etc.)
This is the minimal status that participants are allowed to sign up
for.  Avoid duplicates (however the list is still approximately 3
times what will actually run).

  <LI>Assigned - Session has participants assigned to it.

  <LI> Scheduled - Session is in the schedule (do not set this by hand
as the tool actually sets this for you when you schedule it in a
room!)  The language needs to match what you want to see
<b>published</b>.

</UL>
```

3.5.5 Volunteer_Jobs_0

```
<UL>
<H3>Introducer: (in room)</H3>
  <LI> Sign in at the Green Room, so we know everything is covered.
  <LI> Collect anything needful, like handouts and blank surveys, or
  if it is the first class of the day, the signs, from the Green Room.
  <LI> Be at class 10 minutes early (at the actual end of the previous
  class).
  <LI> You may, if you wish, pre-stage surveys on people's seats for
  when they arrive.
  <LI> At the beginning of class, move to the front of the room and do
  the introduction.  The Con Blurb and the speaker(s) bio(s) as
  provided.
  <UL>
```

```
    <LI>NOTE: A board member or member of the organizing team may step
    forward to do the introduction, in which case, please hand them
    the paper to do it off of.
  </UL>
  <LI> Take the head count of the class (twice) and write them in the
  spots provided on the introduction paper.
  <LI> Be in the back of the room during class so:
  <UL>
     <LI> When the Runner comes to check on the room, you can let them
     know if there is anything needed.
     <LI> If there is vending in the class, you might need to mind the
     table, while the presenter is presenting, if they don’t already
     have someone assisting them.
     <LI> Using the signs provided, give the 10 minute, 5 minute, and
     Done warnings.
     <LI> Hand out/collect surveys and pencils at the end of class.
  </UL>
  <LI> Do the hand-off to the next Introducer, which includes the
  blank surveys and the signs.
  <LI> Return the filled out surveys collected folded in the class
  sheet, when you are checking in at the end of your stint.
<H3>Volunteer: (outside room)</H3>
  <LI> Check that people coming into the room have the correct
  wristbands.  If they do not, politely send them to registration (if
  it is open) to get them.  If there is an issue, notify the point
  person.
  <LI> Stay at the door during class to ensure that excessive ins and
  outs don’t occur.
<H3>Runner: (all over con)</H3>
  <LI> Ensure that every class room has what it needs.
  <LI> Ensure that that A/V and supply needs of a class are met prior
  to it beginning.
  <LI> You can quietly and respectfully bring any supplies into the
  room as a class is going on, and make sure the Introducer knows what
  was delivered.
<H3>Green Room: (green room)</H3>
The green room is a space designated for Presenters, Programming
Volunteers, Panelists, and Assistants only.  While you are welcome to
hang out there, it is also the Programming Team’s Ground Zero, so, you
might be pressed into service.
  <LI> Assist the Program Participants, including disseminating their
  packets as necessary.
  <LI> Check in and out the Introducers and Volunteers as they come on
  and off their stints.
  <LI> Make sure all necessary supplies are available for the
  volunteers as they arrive for check in, including any handouts.
  <LI> Be available to collect the surveys, etc as they arrive.
  <LI> Stay in the room and hang out with everyone!
  <LI> Be in contact with the Programming Point Person for any
  problem.
</UL>
```

3.5.6 Welcome_Letter_Presenters_0

```
<P>Dude!

<P>Thanks for helping us, man.  You really came through.  Like,
everyone learned bags of info, and your flow was rad!

<P>Every hand was good, yours were great!  High-five!

<P>Dude!

<P>The org-folk.
```

3.5.7 Welcome_Letter_Presenters_and_Volunteers_0

```
<P>We would like to express our gratitude for your contribution to the
Ancient Order of the Spies Convention.  Your expertise, wisdom,
experience, and willingness to share your knowledge are critical
elements in what will make our event a success.  We here at Opsidec do
all we can to create a safe and inviting environment for all
secretive/spying/hiding people, but we must also rely on support from
generous allies such as yourself.  Your time and effort are much
appreciated, and are a benefit to all in this lifestyle.

<P>You contribution is helping us to create an event in which any and
all people can learn and access information that they may not have
available to them in their general life.  This process is crucial to
expand knowledge and support throughout our cities and also throughout
the world.  You are assisting in constructing a safe and supportive
atmosphere that truly fosters our community.  Thank you again for your
participation in the Ancient Order of the Spies Convention.  Your
addition to this event is an advantage to all.

<P>With much gratitude,

<P>Opsidec Limited Organizers
```

3.5.8 Welcome_Letter_Volunteers_0

```
<P>We would like to express our gratitude for your contribution to the
Con of your Dreams.  Your willingness to share your time and energy
are critical elements in what will make our event a success.  We here
at Dream Productions do all we can to create a safe and inviting
environment for all sleapers, but we must also rely on support from
generous allies such as yourself.  Your effort is much appreciated,
and is a benefit to all in this lifestyle.

<P>You contribution is helping us to create an event in which any and
all people can learn and access information that they may not have
available to them in their general life.  This process is crucial to
expand knowledge and support throughout our cities and also throughout
the world.  You are assisting in constructing a safe and supportive
atmosphere that truly fosters our community.  Thank you again for your
participation in the Con of your Dreams.  Your addition to this event
is an advantage to all.

<P>With much gratitude,

<P>The Programming Team
```

4 Database setup

You should already have mysql set up. If mysql is not already set up, a good guide to setting up a mysql server is:

`http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch34_:_Basic_MySQL_Configuration`

The pieces of information you will need are from the above decisions for the db_name.php file:

- database hostname: (**DBHOSTNAME**) (If you are setting things up on the same machine that MySQL is running on, *localhost* should be what you are using for **DBHOSTNAME**).
- database username: (**DBUSERNAME**)
- database password: (**DBPASSOWRD**)
- database name: (**DBNAME**)
- Each of the alternate database locators that will be used: (**BIODB**), (**REPORTDB**), (**CONGODB**), (**LIMITDB**), (**LOCALDB**), (**TIMECARDDB**)
- your host name (**MYHOST**) (If you are setting things up on the same machine that MySQL is running on, *localhost* should be what you are using for **MYHOST**)

4.1 Hosted server

If you are going to have your database served from a machine that is running cpanel or some other menu-based software, the method of setting up your database should be documented there.

The chances are your setup will have you:

- create a database or databases
- possibly create a MySQL user
- add MySQL user to the database or databases
- grant the MySQL user all privs.

4.2 Your own MySQL Setup

If you are setting up your own MySQL server, and need to set up the database by hand the following steps should work for you. Don't forget to replace the instances of **DBHOSTNAME**, **DBUSERNAME**, etc. with the proper bits of information.

- Log into the database: (it should ask you for your MySQL root password)

```
mysql -h*DBHOSTNAME* -p -u root
```

- Create your database:

```
create database DBNAME;
```

- Grant **DBUSERNAME** user access with the password of **DBPASSWORD**:

```
grant all on DBNAME.* to 'DBUSERNAME'@'MYHOST' identified by 'DBPASSOWRD';
grant lock tables on DBNAME.* to 'DBUSERNAME'@'MYHOST';
```

- Reset the privilages

```
flush privileges;
```

5 Database populate

change directories until you are in the Install directory, then:

```
mysql -hDBHOSTNAME -p -uDBUSERNAME DBNAME < ./EmptyDbase.dump
```

6 Database tweaks

Some of the tables in the database don't yet have appropriate front-ends, so, to customize them for your particular event, you will need to modify them directly from the MySQL client. As development proceeds, these will get fewer over time.

Currently, they are:

- Divisions:: If you want some other divisions than Other, Programming, Events, Fixed Functions, Hotel, Unspecified, and Volunteer.
- EmailCC:: Needs to be customized for your convention.
- EmailFrom:: Needs to be customized for your convention.
- EmailTo:: Might need to be customized.

- Features:: List of things that can be in a room. Might need to be customized.
- Phases:: The "Phase" you are in will need to be changed as your phase changes.
- PreconHours:: If you are tracking volunteer hours, the PreconHours will probably need to be added to.
- PubStatuses:: Depending on the useage of the software, you might need more statuses than Prog Staff, Public, Do Not Print, and Volunteer.
- QuestionsForSurvey:: You might want to change these.
- RegTypes:: Depending on how you use it, the RegTypes may change.
- Roles:: Fairly standard, but might want to be customized for your convention.
- RoomSets: Fairly standard, but might want to be customized for your convention.
- Rooms:: This definitely wants to be customized for your convention.
- Services:: List of services that can be provided to a room. Might need to be customized.
- SessionStatuses:: Might need to be customized for your convention.
- Tracks:: Probably will want to be customized for your convention
- Types:: May want to be customized for your convention.

Also, some of the Permission interconnects might have to be customized for your convention.

One set of tables that you might be updating across the life of this instance of Zambia is the Reports table. As people generate useful reports, they do tend to get shared. We hope that, should you develop noteworthy reports, you share them back with the community at large, as well.

Loading such reports are often as simple as:

```
mysql -hDBHOSTNAME -p -uDBUSERNAME DBNAME < ./NewReports.sql
```

Sharing them is as simple as, say, exporting your new report called *voltimepanelists*:

```
echo "SELECT * FROM Reports WHERE reportname='voltimepanelists';" |
mysql -hDBHOSTNAME -p -uDBUSERNAME DBNAME > ./NewReports.sql
```

7 Account creation

7.1 Standalone

If you are going to be using Zambia and not some other registration package, you are going to need access to the program, to begin adding the people who are going to be working with the system.

Currently the easiest way to do so is to add the first three users, by pulling in the Initial_Users.sql file from the *Install* directory.

```
mysql -hDBHOSTNAME -p -uDBUSERNAME DBNAME < ./Inital_Users.sql
```

Once you have done that, you can log in to Zambia using the badgeid of **101** and the password of **changeme**.

You then can modify the appropriate information. Under the *Manage Participants & Schedule* tab, there is an *Administer Participants* choice. Selecting that will allow you to update your password (_important step_) and the "Edit Further" link at the bottom of the page will allow you to update the information so it actually matches you.

Feel free to then go and add the rest of your staff, off of the *Enter Participants* link.

7.2 Congo

You might want to complete the activites above, just to make sure you have access, but once you do, you can migrate the congo data into the system, so all the other folks have their information added.

From congo, do:

```
export_program_participants_congo.sql
```

This generates sql that can be, in turn, locaded into Zambia.

7.3 Not Congo

Tying this into another registration system is slightly more complicated. The easiest way is to use the "regtype" field to track the registration number that the various other registration programs give you, and see if there is a way to massage their data into the "CongoDump" format.

8 First steps

8.1 Schedule

Establishing the schedule of activities in the form of a "todo" list is probably the first thing you wish to do.

```
YourWebPath/webpages/genreport.php?reportname=tasklistdisplay
```

Replacing, of course *YourWebPath* with the proper URL to get to your Zambia-FFF branch install.

8.2 Brainstorming

The Brainstorming links should work immediately. From the top directory (index) page of your site, you should be able to click on the "Suggest a Session/Presenter" button and get right into it.

9 Backing up

Under the *scripts* directory there is a nice little shell-script that you can call with cron to back your information up. If you are to use it, make sure you create the *Data_Backup* directory under the *Local* directory before you use it. I back up weekly several months before the con, start in on daily once heavy changes are being made, so we loose less information if there is a problem, and then about a month or so after the con, back off to weekly or monthly. At one point in time, I was running it hourly, just to be sure.

The script is invoked as:

```
backup_mysql /your/path/to/Zambia-FFF/instance
```

10 Quick and dirty

To build a quick and dirty copy on the same machine as another running version:

- for i in ../Zambia-FFF/* ; do ln -s $i . ; done
- rm Local
- rm webpages
- mkdir Local webpages
- cd webpages
- for i in ../../Zambia-FFF/webpages/* ; do ln -s $i . ; done
- cd ../Local
- mkdir Verbiage
- for i in ../../Zambia-FFF/Local/*.html ; do ln -s $i . ; done
- ln -s ../../Zambia-FFF/Local/Participant_Images .

Finally, copy over, and modify the db_name.php as appropraite To update them all do:

```
for i in FFF-[34]*
  do
    pushd $i/webpages
    for j in ../../Zambia-FFF/webpages/*
      do
ln -s $j .
      done
    popd
  done
```

11 Adding a year instance - Con Structure

Another Con instance is approaching. To set up for this, there is one file and a number of table additions you need to make. At some point the table additions might have their own form. Feel free to update them any way you are comfortable with at the moment. The update to the Local/db_name.php file should only be the update to the CON_KEY entry in the file, all the rest should remain the same, and probably that should migrate to something database setable at some point. The rest of the information below should be set in your database tables.

11.1 ConInfo

The first place to start is setting your ConInfo information. This allows for the con to show up in the index file, and opens all the information up for access, addition, and whatever else you want for your con. If you wish to see what values were set for a previous instance of your con (to make sure you have the right values), you might want to use a query (replacing the $conid with the previous con you want to model on) like:

```
SELECT
    *
  FROM
```

```
      ConInfo
  WHERE
    conid=$conid;
```

The values are:

- conid: The unique number of your con.
- conname: The name of your con (probably inclucign either the number, or the year of your con or other unique identifier).
- connumdays: How many days your con will run.
- constartdate: When your con begins, in the format of YYYY-MM-DD hh:mm:ss and preferably (for ease of thought and scheduling) having your hh:mm:ss be 00:00:00 or midnight.
- conurl: This should be the head of your tree, and match all the previous conurls for this con.
- conlogo: This should point to the logo of your con. If you do not modify it event-to-event, it will probably be the same as the previous conlogos for this con.
- condefaultduration: This is the default session length in hh:mm format.
- condurationminutes: This is an enum set to either ’TRUE’ or ’FALSE’ (yes, this is somewhat misleading, and probably should be changed to something else, globally). If it is set to ’TRUE’ then all times will be reported as just minutes. If it is set to ’FALSE’ then all times will be in hh:mm notation where appropriate.
- congridspacer: The default spacer value for the grid, in seconds. This should probably have several entries, in case the grid wants to be different, for different departments, but that is an enhancement, to come.
- conallowkids: This is an enum set to either ’TRUE’ or ’FALSE’ (yes, this is somewhat misleading, and probably should be changed to something else, globally). If it is set to ’TRUE’ it allows for children, and special children programming. If it is set to ’FALSE’ the special children programming will not appear.
- contotalsess: The maximum number of total sessions you will allow your various people to sign up for. This can vary due to the length of the session, how many days your con will run, etc.
- condailysess: The maximum number of sessions on a particular day you will allow your various people to sign up for. This can vary due to the length of the session, how many days your con will run, how many sessions there are in a day, and how hard you are willing to work them.
- conavailabilityrows: How many rows of on/off times your people can put in, for their availability. The longer the con, or the shorter the session, the more rows you want to allow them to have.

11.2 Phase

This also has to be set up so the appropriate Phase shifting can happen. The phase of the convention will shift, over time, as deadlines approach and pass. Not all of the phasetypeids from the PhaseTypes table need to be there, but, it is easier if they are, so all you have to do is change the phasestate, using the AdminPhases.php page and the appropriate phase changes state. If you wish to see what values were set for a previous instance of your con (to make sure you have the right values), you might want to use a query (replacing the $conid with the previous con you want to model on) like:

```
SELECT
    *
```

```
  FROM
      Phase
    JOIN PhaseTypes USING (phasetypeid)
  WHERE
    conid=$conid;
```

The values are:

- phaseid: the unique key for this entry
- conid: The con that these phases are for. This should be set to whatever conid you are creating.
- phasetypeid: This is the phase that is being turned on or off, as described in the PhaseTypes table.
- phasestate: If a phase is active.

You might wish to simply populate your phases with (replacing $conid with the appropriate conid, and the last "," with ";"): INSERT INTO Phase (conid, phasetypeid, phasestate) VALUES

```
SELECT
    concat("($conid,",phasetypeid,",0),") AS ValueSet
  FROM
      PhaseTypes;
```

An example of phase shifting, is when you open your call for presenters, or you close your call for vendors, or you allow feedback. This can change programatically on specific dates around your con, or you can set them by hand, when you are ready for a phase shift. Many phases (or few) might be active at any given point in time.

12 Adding a year instance - People

12.1 UserHasPermissionRole

This need to be set so people can actually log into your con. This defines the access any particular individual might have. Individuals might have several sets of accesses, depending on their roles, some cover more than one area, others do not. This gets somewhat complex, in the interweaving of some other tables, but, theoretcially that was set up, and still should hold true from previous instances of your con. Of course, tweaking from year to year might need to happen. If you wish to see who had which leadership roles in a previous instance of your con (to make sure you are rolling forward with the right roles), you might want to use a query (replacing the $conid with the previous con you want to model on) like:

```
SELECT
    concat(badgeid," - ",pubsname) as Who,
    concat(permroleid," - ",permrolename) as What
  FROM
      UserHasPermissionRole
    JOIN PermissionRoles USING (permroleid)
    JOIN Participants USING (badgeid)
  WHERE
```

```
    conid=$conid AND
    permrolename like "%Super%";
```

This will give you the badgeids and the permroleids to add to the table. It is in the format of:

- badgeid: The unique id of an individual in Zambia.
- permroleid: One of the set of permissions that an individual will have for this con.
- conid: The con that these permission are for. This should be set to whatever conid you are creating.

If you want to only add your particular badgeid, permroleid, conid information, and then use the "Migrate Participant from another con-instance" link under the "Manage Participants & Schedule" tab to give each the right permissions, that will work as well. But at least you will need to be set up, so you can grant other people access. The use of that page is well documented in the Presenter_Flow document in this directory.

You probably also want to add the 100-user (the Brainstorm User called Idea Submissions), to permroleid 4 (Brainstorm) so that the brainstorm submission system will work.

12.2 UserHasConRole

This is very similar to the UserHasPermissionRole, and was split off, because not all Permission Roles want to necessarily be advertized, and sometimes a Permission Role is a superset of a Con Role that actually wants to be called out. While this is not a necessary step, it is a good one to follow, so people know who is responsible for what, when they visit your webpage. If you wish to see who had which leadership roles in a previous instance of your con (to make sure you are rolling forward with the right roles), you might want to use a query (replacing the $conid with the previous con you want to model on) like:

```
SELECT
    concat(badgeid," - ",pubsname) AS Who,
    concat(conroleid," - ",conrolename) AS What
  FROM
      UserHasConRole
    JOIN Participants USING (badgeid)
    JOIN ConRoles USING (conroleid)
  WHERE
    conid=$conid;
```

This will give you the badgeids and conroleids to add to the table. It is in the format of:

- badgeid: The unique id of an individual in Zambia.
- conroleid: A role that an individual will have for this con.
- conid: The con that the role is for. This should be set to whatever conid you are creating.

You can always go back and add to this, as the shape of your con fills in.

If you want to only add your particular badgeid, permroleid, conid information, and then use the "Migrate Participant from another con-instance" link under the "Manage Participants & Schedule" tab to give each the right permissions, that will work as well. But at least you will need to be set up, so you can grant other people access. The use of that page is well documented in the Presenter_Flow document in this directory.

You probably want to add the 100-user (the Brainstorm User called Idea Submissions), to conroleid 47 (BrainstormCoord) for tracking purposes.

12.3 HasReports

This maps the organizational chart of your convention. Again while this is not a necessary step for the start of your con, and might be added to as your con goes on using the AdminHasReports.php page, usually you know what the table of your organization will be, even if you don't yet know who will be filling those roles. To see what was before you can use a query (replacing the $conid with the previous con you want to model on) like:

```
SELECT
    concat(HR.conroleid," - ",CR.conrolename) AS Role,
    concat(hasreport," - ",CRR.conrolename) AS Report
  FROM
      HasReports HR
    JOIN ConRoles CR USING (conroleid)
    JOIN ConRoles CRR ON (hasreport=CRR.conroleid)
  WHERE
    conid=$conid;
```

This will give you the list of conroleids who have conroleids reporting to them. You can add them to the table as:

- conid: The con that these reports are for. This should be set to whatever conid you are creating.
- conroleid: This is the role that has others reporting to them
- hasreport: The conroleid of the position that reports to the specified original conroleid.

On the other hand, if you just want to mirror them, quickly, from another instance of your con you could always (replacing $origconid with the previous con you want to model on, and $conid with the current one) run:

```
SELECT
    concat("($conid,",conroleid,",",hasreport,"),") AS Prev
  FROM
      HasReports
  WHERE
    conid=$origconid;
```

Then just massage the data slightly, and you have the insert you can do to mirror the previous con.

You probably want to make sure the BrainstormCoordinator (47), reports to Programming (3).

13 Adding a year instance - Sessions

The sessions are mostly generated from Brainstorm (and other insertions to your database) but there are a few settings that should be taken care of, to make sure everything works smoothly. The limits are set on an event basis, since sometimes publications or the space for things to be advertized differs from year to year. Also the requestables might change, depending on what is available, as well as the rooms

13.1 PublicationLimits

This tracks the various limits on the publication fields, so that you can have restrictions on minimum and/or maximum length of these fields. All the limit checks should be set up such that if you don’t have a limit set in this table, the minimum is presumed 0 and the maximum is presumed to be whatever your database maximum is for that particular field. If you wish to see what limits were set for a previous instance of your con (to make sure you have the right limits), you might want to use a query (replacing the $conid with the previous con you want to model on) like:

```
SELECT
    *
  FROM
      PublicationLimits
  WHERE
    conid=$conid;
```

The values are:

- publimid: Unique key for this table (it should auto-increment).
- conid: The con that these limits are for. This should be set to whatever conid you are creating.
- publimtype: This is an enum between two values, min and max. This determiines if the limit is a minimum limit, or maximum limit.
- publimdest: This is an enum between web and book, and should be in sync with your BioDests table. This is so you can set different limits for the various publication media.
- publimname: This should be in sync with your BioTypes table, and is the specific field that is being limited.
- publimval: The number of characters you are limiting this type of entry to.
- publimnote: Any notes you want to have about the limit.

On the other hand, if you just want to mirror them, quickly, from another instance of your con you could always (replacing $origconid with the previous con you want to model on, and $conid with the current one) run:

```
SELECT
    concat("($conid,’",publimtype,"’,’",publimdest,"’,’",publimname,"’,",publimval,",’",publimn
  FROM
      PublicationLimits
  WHERE
    conid=$origconid;
```

Then just massage the data slightly, and you have the insert you can do to mirror the previous con.

13.2 Services and Features

These two tables are built in the same pattern. Services are the bits that logistics could get to the room, so they could be used for the class. Some of the room features (speakers, microphone) fall under this, because of being extra cost. Features are elements of the room that someone would look for. With a single room, there might not be much there to choose between, but, with many rooms, or different room functions depending on the time of day (say, needing curtains in the daytime, and dimable lights, so that projection will work correctly) so scheduling can happen more easily.

To select (or update the selection) of services and features available for this con-instance, use the AdminSetupServices.php page.

13.3 Rooms

This needs to be fleshed out more.

14 Adding a year instance - Vendors

14.1 VendorSpaces and VendorFeatures

These two tables are built in the same pattern. VendorSpaces are the size/type of space, and the cost of said space for the con. VendorFeatures are all the little up-charges or non-standard things that a vendor might opt for. To see what was set on a previous instance of your con, you can use a query (replacing the $conid with the previous con you want to model on) like:

```
SELECT
    *
  FROM
      VendorSpaces
  WHERE
    conid=$conid;
```

And:

```
SELECT
    *
  FROM
      VendorFeatures
  WHERE
    conid=$conid;
```

The values are:

- vendorspaceid/vendorfeatureid: Unique id for the entry (key).
- vendorspacename/vendorfeaturename: A description, plus the price in an easily understandable format, so the vendors know what to select for their comfort and profit.
- vendorspaceprice/vendorfeatureprice: Numeric representation of the cost of said item, so the invoice can be generated correctly.
- conid: The con that these things are available for . This should be set to whatever conid you are creating.
- display_order: For the select boxes, if you want to reorder possibilities, this forces the ordering to what you want it to be.

On the other hand, if you just want to mirror them, quickly, from another instance of your con you could always (replacing $origconid with the previous con you want to model on, and $conid with the current one) run:

```
SELECT
    concat("('",vendorspacename,"',",vendorspaceprice,",$conid,",display_order,"),") AS Prev
  FROM
      VendorSpaces
  WHERE
    conid=$origconid;
```

And:

```
SELECT
    concat("('",vendorfeaturename,"',",vendorfeatureprice,",$conid,",display_order,"),") AS Pre
  FROM
      VendorFeatures
  WHERE
    conid=$origconid;
```

Then just massage the data slightly, and you have the insert you can do to mirror the previous con.

14.2 More Vendor stuff, pending development.

15 Adding a year instance - Feedback

Setting up feedback for your con is somewhat complex. I'm not sure what motivated me, originally to be so baroque, but I think it was flexibility at the time.

15.1 Adding Questions

If your questions are not appropriate for this particular event, you might want to add a question type to cover it, in the QuestionTypes table, and then add the questions to the QuestionsForSurvey table.

The values for QuestionTypes are:

- questiontypeid: Unique key for this table (it should auto-increment).
- questiontypename: A one or two word name for the type of question.
- questiontypenotes: A more indepth description of the question type.

The values for the QuestionsForSurvey are:

- questionid: Unique key for this table (it should auto-increment).
- questiontext: The text of the question to be asked.
- display_order: The order of the question to be asked.
- questiontypeid: What type of question it is, referencing the QuestionTypes table above.

15.2 Setting up the page description (how it is presented)

For your event, there might be several different feedback forms you want available, so that an individual isn't overwhelmed by the number of possible choices. One of the ways to do this is to break it down by time period, another is to break it down by class type. Once you have broken them down, this table sets up the various possible views. At least one view should be set to "single class", so that the reference from the descriptions etc will work.

The values for FeedbackPages are:

- fpageid: Unique key for this table (it should auto-increment).
- conid: For which con is this page relevant.
- fppagedesc: The short description of the page contents.
- fpagestart: The start time of the schedule elements that this page covers.
- fpageend: The end time of the schedule elements that this page covers.
- fpagecols: The number of columns at the top of the printed page.
- questiontypeid: Which type of questions will show up on this page, from the QuestionTypes table.

The type of schedule elements included in each possible Feedback page is set in the FeedbackPageHasType. This can have multiple mappings. For example, if you have both classes and panels, they have different values for their typeid in the Sessions table. To have them both show up in the "single class" page, that page id needs two entries in the FeedbackPageHasType table, each one mapping the typeid to that fpageid. "single class" should have a mapping for every element you reference, but the feedback page for just the classes would only have the typeid of classes mapped to it.

The values for FeedbackPageHasType are:

- fpageid: From the FeedbackPages table.
- typeid: From the Types table.

15.3 Setting the TypeHasQuestionType

This is implemented to be able to cross-type questions, and make sure you wanted those question types to show up for that type of schedule element, for this particular event. It does allow for some greater flexibility, as opposed to having a fixed QuestionType for the schedule Type, since that might change from event to event. Smaller events might want all the schedule Types to have the same QuestionType, larger or more siloed events might want to break it down further. Depending on what is being fed back, Divisions might creep into this, as well.

The values for TypeHasQuestionType are:

- typeid: From the Types table.
- questiontypeid: From the QuestionTypes table.
- conid: For which event is being run.