

SciCon^{PHP} Technical Document

Denice C. Deatrach

May 5, 2002

Last update: Oct 12, 2003

`$Id: sciconphp-technical.lyx,v 1.8 2003/10/12 16:01:12 deatrach Exp $`

Contents

| | |
|--|-----------|
| 1. Introduction | 3 |
| 1.1. Conventions Used in This Document | 3 |
| 2. Programming Conventions | 4 |
| 2.1. File and Function Header Formatting | 4 |
| 2.2. Coding Style | 5 |
| 3. Program Design | 5 |
| 4. Important Files and Their Role | 6 |
| 4.1. Typical Web Page Composition | 6 |
| 4.2. Other Noteworthy Files | 7 |
| 5. Database Design | 7 |
| 5.1. Specific Database Table Issues | 7 |
| 5.1.1. The PEOPLE Table | 7 |
| 5.1.2. The PAPER Table | 8 |
| 5.1.3. The AUTHORLIST Table | 8 |
| 6. Database Table Descriptions | 8 |
| 7. Outstanding Issues, Bugs and Other Goodies | 13 |
| 7.1. List of issues to address | 13 |
| 7.2. Bugs | 14 |
| 7.3. Useful things yet to do | 14 |
| 8. Appendix | 14 |
| A. topics to cover | 14 |

1. Introduction

Installation and configuration issues are documented in the SciCon^{PHP} Installation document. This document focuses on design and program issues for people who need to understand the underlying structure and logic (or the lack thereof).

In this document I will try to steer you through the code, so that you can:

- understand enough about the code to be comfortable editing the general webpage content.
- understand the database relations
- find and fix existing bugs.

There are a few bugs if you are using an early release of this code; there will probably still be bugs in future releases. In particular I have revamped the code to get rid of the need for the PHP directive *register_globals* to be turned on. This has introduced some bugs.

1.1. Conventions Used in This Document

- Areas of the documentation that are incomplete for one reason or another are marked thusly:

(to be completed)

- pathnames are shown in `/fixed/font/`
- commands, names and empasized items *are shown in italics*
- things that should be noticed are **sometimes colored red**
- comments in configuration file examples in the appendices are **colored blue** while the directives or configuration items are left in black, **but the font is fixed and small** in these examples. Again, items which wish to **draw your attention are colored red.**
- when working at the command-line as an ordinary user the command prompt is a `$` and as user root the command prompt is a `#`

```
$ some_command as_an_ordinary_user
# some_command as_the_superuser
```

2. Programming Conventions

2.1. File and Function Header Formatting

All files and functions in the so-called 'outside' directory contain comment headers like the following examples.

This is an example of a file header. It contains a CVS Id tag, a description of the file's purpose, and a reiteration of the GPL license text:

```
/**
 * FILE:      i_submissionFunctions.php
 * DESCRIP:   these are all the functions to include with the paper
 *            submissions pages:  sub*.php
 *
 * @version   Id:  i_submissionFunctions.php,v 0.9.0.1 2002/04/26 15:51:40 deatrich Exp $
 * @copyright This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License as published
 * by the Free Software Foundation; either version 2, or (at your option) any
 * later version.
 *
 * This program is distributed in the hope that it will be useful, but
 * WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * General Public License for more details.
 */
```

Both file and function headers use a javadoc-like syntax so that they can be parsed by phpDocumentor

This is an example of a function header. Generally used phpdoc tags for functions include @param and @return. See the complete list of tags and their formats in the phpdoc online documentation.

```
/**
 * PURPOSE:   authenticate the user before allowing him/her access to pages
 * NOTES:     will exit on error conditions
 * @param     string $email -- the user's email addr
 * @param     string $pw1 -- the user's password
 * @param     boolean $checkPwd -- whether to check the password
 * @param     array $a -- the TPC member array to fill in this function
 * @return    n/a
 */
function authenticateReviewer( $email, $pw1, $checkPwd, &$a )
{
    ....
}
```

2.2. Coding Style

Here are some general notes about the coding style:

- The coding bracketing style is essentially Kernihan and Richie.
- Indentation tends to be 2 or 3 spaces. I tend to make function calls look 'airy' with lots of spaces within the argument list. This drives some people crazy .. sorry..
- Functions tend to be 'single-entry, single-exit', except of course when they exit completely and totally on an error condition. I hate 'return' statements occuring before the end of functions.
- There are one or two naming conventions; unfortunately there are not enough:
 - ◊ Files in the `/path/to/outside/inc/` directory have a naming prefix *i* – for example `i_commonFunctions.php`
 - ◊ database function names start with *db* –
- Any system error, and most database errors are coded as exits, using this function found in `i_commonFunctions.php`
`exitWithError($msg, 0);`

Indeed, the code makes extensive use of error checking, particularly for database and, of course, system errors.

3. Program Design

The first thing that you will notice when you browse through the PHP source code is that it doesn't look very object-oriented. That's right. It looks rather like other procedural languages like *C*. Of course, *PHP* is extremely flexible, and allows for widely different implementation approaches. That's one of the reasons it is so popular.

The code lives in two areas:

1. The so-called *inside* tree which resides within the apache-served document root.

The pages provided by the code for this tree are mostly skeletal. You will need to edit them and provide your own content. You do not need to know much PHP, but you will have to know some HTML. You might find an intelligent web editor for this task, but you will inevitably have to know a fair bit of HTML to be comfortable in the PHP world. There are some pages in the

reviewer and the admin areas which are very 'busy'. In future releases much of the code in these pages should be moved to the outside tree.

2. The *outside* tree, which of course, lives outside of the document root.

These are the library and include files that are the heart of SciCon^{PHP}.

4. Important Files and Their Role

4.1. Typical Web Page Composition

All pages are composed of four or five parts:

1. An HTML header
2. Perhaps some javascript. Javascript inclusion is controlled by the variable *\$includeJavaScript*
3. Perhaps a sidebar menu. Menu inclusion is invoked by the function *insertMenu()*
4. the actual page contents
5. And finally (usually) a page footer. Footer inclusion is controlled by parameters to the function *insertFooter()*

| File and/or Function | Description |
|---|---|
| i_header.php | The head part of the page, which includes the code for the menu and the footer functions mentioned below. |
| i_js.php | The javascript code, which is conditionally included. |
| insertMenu() i_header.php i_menu*.php | The side bar menu (implemented as a multi-dimensional array, which can thus be easily modified), called as a function: insertMenu(\$header, [\$includeMenuFileName],) |
| example.php | The page contents; this is just an example: <pre><?php include("i_header.php"); insertMenu("This is the header for this example"); echo " <p class=\"warn\"> This is an example page that has no real content!</p>"; insertFooter(1); ?></pre> <p>the generated page for this example code looks like this [provide link].</p> |
| insertFooter() i_header.php | The end of the page, or page footer – it is called as a function: insertFooter(\$flag,) |

4.2. Other Noteworthy Files

Other files which aid in generalising the site content and/or design are:

| File | Description |
|---|---|
| i_definesSite.php | The site definitions are all in this file, so that the pages can be easily changed to host another conference. |
| i_defines.php | Common definitions for the PHP functions. The context-sensitive HELP is also in this file. |
| style.css | The CSS style sheet; it can be modified to completely change the appearance of the site. |
| i_mysql.php | The database functions separate from the code (in case one wants to switch database technologies). |
| i_dbConnect.php | The database authentication file, separate for obfuscation reasons. |
| i_affil.inc, i_category.inc, i_country.inc, i_pfx.inc, i_sfx.inc, i_sessions.inc, i_initials.inc | These are simple text files each containing a relevant list of items separated by line breaks. A file is read each time a drop-down list of this type is needed in a form. The text files on the left correspond to (in order) – affiliations, paper submission subject categories, countries, name prefixes, name suffixes, final paper presentation session names and initials of the registration admin who processed a payment. |

Except for a few functions found in i_header.php, all other functions are found only in the following include files:

- i_mysql.php, i_*Functions.php

5. Database Design

The following tables describe the columns and data types of the tables. Primary keys are colored yellow; ancilliary unique constraints are colored pink in the PDF and PostScript renditions of this document.

NOTE: Early in program development the database design was changed, and there are still some remnants of this change in the database. Specifically the paper and the authorlist tables contain a few duplicated fields which need to be removed in a subsequent release.

5.1. Specific Database Table Issues

5.1.1. The PEOPLE Table

In principle the PEOPLE table could use the email address as a unique key. Unfortunately for international conferences there are still authors who do not

have an email address. Indeed the IEEE Information Theory conference policy in 2002 still allowed offline paper submissions (good old typewritten papers delivered by the post office). This may still be the case - I have not checked recently. Moreover during the last few days of paper submission prospective paper contact authors may not have the email addresses of everyone in the authorlist, and they will just enter in any old thing to get their paper submitted before the deadline. And some contact authors are just lazy and hate web forms. For the ISIT conference we tried to make life as easy as possible for the contact authors and we minimized the amount of data entry required to submit a paper. In retrospect this wasn't such a good idea, but you will see its effects in the code: there is an email address defined as: **NOMAIL**.

Of course, the email address has other problems: people typically have more than one address, and some seem to use them randomly. Other people change email addresses almost as frequently as they change their clothes. Thus the combination of last,first,suffix is in principle more stable. It is not necessarily unique though. In short, its an imperfect world, and it is reflected in the code and the table definitions...

5.1.2. The PAPER Table

The *email* field should be jettisoned from this table and the code in a future release. It is an unnecessary duplication of information provided by the *contact* field.

5.1.3. The AUTHORLIST Table

The *contact* field should be jettisoned from this table and the code in a future release. It is an unnecessary duplication of information provided by the *contact* field in the paper table.

6. Database Table Descriptions

PEOPLE data describing any participant (except a paper reviewer, who has a slightly different table)

6 DATABASE TABLE DESCRIPTIONS

| PEOPLE | | |
|-----------------------------|------------------------|--------------------------------|
| Column | Description | Data Type |
| id | person number | int(8) unsigned auto_increment |
| last | last name | varchar(30) NOT NULL |
| first | first name | varchar(30) NOT NULL |
| px | optional prefix | varchar(10) default NULL |
| sfx | optional suffix | varchar(10) default NULL |
| affil | affiliation | varchar(80) NOT NULL |
| email | email address | varchar(50) NOT NULL |
| tel | telephone number | varchar(20) default NULL |
| fax | FAX number | varchar(20) default NULL |
| addr | street postal addresss | varchar(80) default NULL |
| city | city name | varchar(30) default NULL |
| prov | province or state | varchar(20) default NULL |
| pcode | postal code | varchar(10) default NULL |
| country | country name | varchar(30) default NULL |
| UNIQUE KEY (last,first,sfx) | | |

PAPER data describing a paper submitted for possible presentation

| PAPER | | |
|-----------|--|------------------------------------|
| Column | Description | Data Type |
| pn | paper number | int(8) unsigned auto_increment |
| title | paper title | varchar(255) NOT NULL |
| numauth | number of authors | int(2) NOT NULL |
| category | paper category | varchar(60) NOT NULL |
| category2 | optional 2nd paper category | varchar(60) default NULL |
| filename | full path to downloaded paper | varchar(255) NOT NULL |
| filesize | size of this file in bytes | int(10) unsigned NOT NULL |
| filetype | file type may be: PDF or PostScript | varchar(50) NOT NULL |
| status | paper's current status | set('PRE','CRE','MOD','DEL','LCK') |
| contact | contact author id # from people table | int(8) unsigned NOT NULL |
| email | contact author email address | varchar(50) NOT NULL |
| pass | author password to access this paper | varchar(16) NOT NULL |
| moddate | date of modification of this row entry | timestamp(14) NOT NULL |
| regdate | date of creation of this row entry | timestamp(14) NOT NULL |

AUTHORLIST data associating people with paper

| AUTHORLIST | | |
|------------|------------------------|--------------------------------|
| Column | Description | Data Type |
| pn | paper number | int(8) unsigned |
| id | person number | int(8) unsigned auto_increment |
| pos | position in authorlist | unsigned NOT NULL |
| contact | 'Y' if contact author | set('Y','N') NOT NULL |

REVIEWER a reviewer who is given a list of papers to review

| REVIEWER | | |
|-----------------------------|---------------------------|--------------------------------|
| Column | Description | Data Type |
| rid | reviewer number | int(8) unsigned auto_increment |
| last | last name | varchar(30) NOT NULL |
| first | first name | varchar(30) NOT NULL |
| px | optional prefix | varchar(10) default NULL |
| sfx | optional suffix | varchar(10) default NULL |
| affil | affiliation | varchar(80) NOT NULL |
| email | email address | varchar(50) NOT NULL |
| pass | password for web access | varchar(16) NOT NULL |
| cochair | if person is a cochair | set('Y','N') NOT NULL |
| dom1 | primary area of expertise | varchar(60) NOT NULL |
| dom2 | 2nd domain of expertise | varchar(60) NOT NULL |
| dom3 | 3rd domain of expertise | varchar(60) NOT NULL |
| UNIQUE KEY (email) | | |
| UNIQUE KEY (last,first,sfx) | | |

REVPAPER data describing the review data for a submitted paper

| REVPAPER | | |
|----------|-------------------------------|--|
| Column | Description | Data Type |
| pn | paper # from paper tab | int(8) unsigned |
| rid | reviewer # from reviewer tab | int(8) unsigned |
| status | paper's current review status | set('ASSN','REVD','ACC','REJ','WTDR') NOT NULL |
| moddate | date of modification | timestamp(14) NOT NULL |
| assndate | date of creation | timestamp(14) NOT NULL |
| cat1 | 1st review category | int(1) unsigned NOT NULL |
| cat2 | 2nd review category | int(1) unsigned NOT NULL |
| cat3 | 3rd review category | int(1) unsigned NOT NULL |
| cat4 | 4th review category | int(1) unsigned NOT NULL |
| cat5 | 5th review category | int(1) unsigned NOT NULL |
| rec | reviewer recommendation | int(1) unsigned NOT NULL |
| self | reviewer confidence | int(1) unsigned NOT NULL |
| award1 | recommend for award 1 | set('Y','N') |
| award2 | recommend for award 2 | set('Y','N') |
| session1 | 1st recommended session | varchar(80) |
| session2 | 2nd recommended session | varchar(80) |
| authcomm | public comments for author | text |
| privcomm | private comments for chair | text |

REVQUALITY paper review qualities for each sub-category (cat1, cat2, ..., rec, self)

6 DATABASE TABLE DESCRIPTIONS

| REVQUALITY | | |
|------------|----------------------|------------------------------------|
| Column | Description | Data Type |
| quality | column from revpaper | varchar(10) NOT NULL |
| type | type of quality | set('specific','general') NOT NULL |
| descr | descriptive phrase | varchar(50) NOT NULL |

REVPOINTS numeric value for each possible selected review quality for each sub-category (cat1, cat2, ..., rec, self)

| REVPOINTS | | |
|-----------|--------------------------------|--------------------------|
| Column | Description | Data Type |
| quality | column from revquality | varchar(10) NOT NULL |
| score | this numeric score | int(1) unsigned NOT NULL |
| meaning | describe what this score means | varchar(50) NOT NULL |

FINALPAPER list of accepted papers and their upload data (for publication in the proceedings)

| FINALPAPER | | |
|------------|-------------------------------|--|
| Column | Description | Data Type |
| pn | paper # from paper tab | int(8) unsigned |
| filename | full path to downloaded paper | varchar(255) NOT NULL |
| filesize | size of this file in bytes | int(10) unsigned NOT NULL |
| filetype | file type : PDF or PostScript | varchar(50) NOT NULL |
| status | paper's current status | set('PRE','CRE','MOD','WTDR','LCK') NOT NULL |
| moddate | date of modification | timestamp(14) NOT NULL |
| regdate | date of creation | timestamp(14) NOT NULL |

SESSIONS data describing the list of all presentation sessions for accepted papers

| SESSIONS | | |
|----------|------------------------------------|--------------------------------|
| Column | Description | Data Type |
| sid | session # | int(8) unsigned auto_increment |
| name | session name | varchar(80) NOT NULL |
| room | name of room for this session | varchar(50) NOT NULL |
| chair | person id # who will chair session | int(8) unsigned NOT NULL |
| timeslot | unique timeslot number | int(4) unsigned NOT NULL |

SESSIONLIST association between sessions and accepted papers for each session

| SESSIONLIST | | |
|--------------------|--------------------------------|--------------------------|
| Column | Description | Data Type |
| sid | session # | int(8) unsigned |
| pn | paper # from paper tab | int(8) unsigned |
| pos | position # of paper in session | int(2) unsigned NOT NULL |

SESSIONTIMES association between timeslots for each paper in a session and the time

| SESSIONTIMES | | |
|---------------------|--------------------------------|--------------------------|
| Column | Description | Data Type |
| timeslot | unique timeslot number | int(4) unsigned NOT NULL |
| pos | position # of paper in session | int(2) unsigned NOT NULL |
| time | time of presentation | timestamp(14) NOT NULL |

CHAIRS complete list of chairpersons for sessions

| CHAIRS | | |
|---------------|---------------|--------------------------|
| Column | Description | Data Type |
| id | person number | int(8) unsigned NOT NULL |

ATTENDEE list of all persons registered for the conference and their payment data and status

| ATTENDEE | | |
|-------------|---------------------------|--|
| Column | Description | Data Type |
| id | person number | int(8) unsigned |
| moddate | date of modification | timestamp(14) NOT NULL |
| regdate | date of registration | timestamp(14) NOT NULL |
| debitdate | date of actual payment | timestamp(14) NOT NULL |
| skey | session key | varchar(50) default NULL |
| skeyexpdate | timestamp of skey use | timestamp(14) NOT NULL |
| status | registration status | set('PRE','ONL','OFFL','DBT','DEL') NOT NULL |
| paytype | payment type | set('CC','FAXCC','CHEQ','WTX','MISC') NOT NULL |
| cctype | credit card type | set('NONE','VISA','MasterCard') NOT NULL |
| ccnum | credit card number | varchar(19) |
| ccmon | CC month of expiry | int(2) |
| ccyr | CC year of expiry | int(4) |
| ccholder | Card holder's name | varchar(50) |
| ccauthcode | CC authorization code | varchar(10) |
| regcat | registration category | varchar(10) |
| opt1 | # of extra tickets opt. 1 | int(2) unsigned NOT NULL |
| opt2 | # of extra tickets opt. 2 | int(2) unsigned NOT NULL |
| billed | amount billed | float(6,2) NOT NULL |
| paid | amount paid | float(6,2) NOT NULL |
| init | payment admin's initials | varchar(5) |
| memnumber | IEEE membership # | varchar(10) |
| comments | payment adm's comments | text |

REGCAT reference list of registration categories and registration options

| REGCAT | | |
|---------|-------------------------------------|-----------------------|
| Column | Description | Data Type |
| cat | column from revpaper | varchar(10) NOT NULL |
| descr | descriptive phrase for category | varchar(50) NOT NULL |
| cost1 | cost for category before deadline | float(5,2) NOT NULL |
| cost2 | cost for category if after deadline | float(5,2) NOT NULL |
| neednum | if category requires membership | set('Y','N') NOT NULL |

7. Outstanding Issues, Bugs and Other Goodies

7.1. List of issues to address

(to be completed)

- postscript editing and the site logo

- the joy of pdf and printing (I'm being sarcastic here - it can be absolutely horrible)

7.2. Bugs

(to be completed)

- some php email-address checking is not entirely correct. This needs to be fixed. In particular some new domains have been introduced. Also, email addresses may contain strange characters like '+', and the functions doing the checks erroneously rejects these addresses.

7.3. Useful things yet to do

- test suite

(to be completed)

8. Appendix

A. topics to cover

- -