

TeamSCOPE Internals

Ben Pfaff

Copyright © 1999 Michigan State University <http://www.msu.edu>

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Free Software Foundation.

1 Introduction

This manual documents the internals of TeamSCOPE, Team Software for Collaborative Project Environments. It is written for C programmers and database administrators. A basic familiarity with TeamSCOPE usage, as well as competence in ANSI C89, Java, and ANSI SQL92, is assumed.

This manual was last updated for TeamSCOPE 0.9.21.

TeamSCOPE and its documentation were written by Ben Pfaff pfaffben@msu.edu.

1.1 Architecture

The TeamSCOPE system contains four programs:

`'scope.cgi'`

Main program, written in ANSI C89, called by the web server. Compliant with the CGI specification (<http://www.ast.cam.ac.uk/~drtr/cgi-spec.html>).

See section “Introduction” in *The TeamSCOPE Users Guide*, for more details on using TeamSCOPE’s web interface.

`'scope-db'`

Utility, written in ANSI C89, used for installation, maintenance, and backup.

See section “`scope-db`” in *The TeamSCOPE Installation Guide*, for more details on `'scope-db'`.

`'scope-chat'`

Chat server, written in ANSI C89, for use with the Java chat client below.

`'ScopeChat'`

Chat client, written in Java, which is embedded in TeamSCOPE web pages automatically when two or more users on a team are both using the system at once.

TeamSCOPE also makes use of programs which it does not contain. In the table below, the generic names of these programs are given, along with the specific product that TeamSCOPE has been tested with and a description of how TeamSCOPE uses the program.

database server (PostgreSQL)

Most TeamSCOPE data is kept in a relational database.

web server (Apache)

Needed for a web server alias and cgi-bin execution rights. TeamSCOPE also creates a per-team web server alias.

ftp server (ProFTPD)

File accesses should be logged, even if they come from outside the TeamSCOPE system. TeamSCOPE includes patches for a modified version of ProFTPD that logs FTP accesses to the TeamSCOPE database.

mail server (Exim)

For research purposes, TeamSCOPE can instruct the mail server to change the Reply-To: headers of team members.

TeamSCOPE makes use of libgd to draw GIF graphics at runtime, using a patent-free run-length encoding compression algorithm.

2 Database Schema

The TeamSCOPE database contains the following tables. For each table, its purpose and the type and purpose of each of its columns is described.

See the source code to ‘`scope-db`’ to see the exact database schema.

2.1 Table: *activities*

Tracks changes in the filesystem, calendar, and message boards. Each row represents one activity.

date (datetime)

Date and time activity occurred.

s1 (int4) Reference to *strings*. See *type* below for meaning. If not used for a particular activity, must be set to 0.

usr (varchar(8))

Reference to *users*. User who performed activity.

grp (varchar(8))

Reference to *groups*. Group in which user performed activity (users may belong to multiple groups).

cat (int2) Category of activity. One of the following:

AC_CAT_FILE (1)

Files and file-related message activities.

AC_CAT_MSG_BOARD (2)

Message board activities.

AC_CAT_EVENT (3)

Calendar activities.

type (int2)

Activity type within a category. For file activities, these are:

AC_FILE_DOWNLOAD (1)

File download. *s1* specifies the downloaded file.

AC_FILE_DOWNLOAD_ARCHIVE (2)

Directory download as an archive, in ‘.tar.gz’ or ‘.zip’ format. *s1* specifies the directory downloaded.

AC_FILE_UPLOAD (3)

File upload. *s1* specifies the file.

AC_FILE_DELETE (4)

File deletion. *s1* is the file.

AC_FILE_COPY (5)

File copy. *s1* is the destination file, *s2* is the source file.

AC_FILE_RENAME (6)

File rename. *s1* is the new filename, *s2* is the original filename.

AC_FILE_MKDIR (7)

Make new directory. *s1* is the name of the new directory.

AC_FILE_RMDIR (8)

Remove empty directory. *s1* is the name of the removed directory.

AC_FILE_CHMOD (9)

Change file permissions. *s1* is the name of the file whose permissions were changed.

AC_FILE_MOVE (10)

Move file between directories. *s1* is the new filename, *s2* is the original filename.

AC_FILE_POST (11)

Post a message about a file. *s1* is the filename of the file commented on, *s2* is the message title, *ref_seq* is the message posted.

AC_FILE_POST_DELETE (12)

Post about a file deleted by poster. Variables as AC_FILE_POST above.

AC_FILE_POST_READ (13)

Post about a file read by other user. Variables as AC_FILE_READ above.

AC_FILE_POST_MAILED (14)

User was mailed article. Variables as AC_FILE_READ above.

For message board activities, the valid values are:

AC_MSG_POST (1)

Message board posting. *s1* is '*//workgroup*', where *workgroup* is the poster's workgroup, for historical reasons. *s2* is the message title, and *ref_seq* is the message posted.

AC_MSG_DELETE (2)

Message board post deleted by poster. Variables are as described in AC_MSG_POST above.

AC_MSG_READ (3)

Message board post read by other user. Variables are as described in AC_MSG_POST above.

AC_MSG_MAILED (4)

Message board post mailed to user. Variables are as described in AC_MSG_POST above.

s2 (int4) Reference to *strings*. See *type* above for meaning. If not used for a particular activity, must be set to 0.

ref_seq (int4)

Reference sequence number. See *type* above for meaning. If not used for a particular activity, must be set to 0 or null.

seq (int4) (primary key)

Sequence number obtained from global sequence generator *seq*.

2.2 Table: *catchup*

This table records which messages the user has decided don't need to be shown to him again on a per-file basis. Records are updated when the user clicks on a message board 'Reset' button.

usr (varchar(8))

Reference to *users*. User in question.

grp (varchar(8))

Reference to *groups*. Group in question.

file (int4) Reference to *strings*. Name of the file that the message applies to. For posts in the global message board, this is '//workgroup', where *workgroup* is the name of the user's workgroup.

last_read (int4)

Reference to *notes*. Last message shown to the user for this file.

seq (int4) (primary key)

Sequence number obtained from global sequence generator *seq*.

In this table, (*usr*, *grp*, *file*) form a unique tuple.

2.3 Table: *conflicts*

This table tracks resource conflicts between calendar events in table *events*. Its use is currently not fully defined since resource reservation is only partially implemented at the moment.

seq1 (int4)

Reference to *events*. First conflicting event.

seq2 (int4)

Reference to *events*. Second conflicting event. *seq2* must be greater than *seq1*.

reported (bool)

Set to true when the conflict has been reported to the user.

seq (int4) (primary key)

Sequence number obtained from global sequence generator *seq*.

In this table, (*seq1*, *seq2*) is a unique tuple.

2.4 Table: *event_resource*

This table associates events with resources reserved for the event duration.

event (int4)

Reference to *events*.

resource (int4)

Reference to *resources*.

seq (int4) (primary key)

Sequence number obtained from global sequence generator *seq*.

In this table, (*event*, *resource*) is a unique tuple.

2.5 Table: *events*

This table describes calendar events scheduled by each team.

start (datetime)

Event start date and time.

finish (datetime)

Event end date and time.

title (int4)

Reference to *strings*. Event title.

note (text)

Any notes about event.

owner (varchar(8))

Event “owner.” Must be username of a user in *grp* below, null, or the empty string.

grp (varchar(8))

Reference to *groups*. Group event applies to.

seq (int4) (primary key)

Sequence number obtained from global sequence generator *seq*.

source (int4)

Source of event. Currently, set to 0 for events entered via web form, 1 for events obtained via MS Project file import.

2.6 Table: *groups*

This table records information about TeamSCOPE workgroups.

grp (varchar(8)) (primary key)

UNIX group name.

name (text)

Long form of group name for presentation purposes.

sharedir (text)

Name of shared directory for group. Must begin with '/', may not end with '/'.

webdir (text)

Name of web directory for group. Must begin with '/', may not end with '/'.
Normally this is '*sharedir/www*'.

md5 (text[])

Used to save md5sums of group files in order to tell whether they have been modified. (Not currently implemented.)

2.7 Table: *notes*

This table contains all message board postings, whether they are associated with a file or not.

file (int4) Reference to *strings*. Associated file for comments on files, or '//*workgroup*', where *workgroup* is the workgroup name, for posts to the global message board.

grp (varchar(8))

Reference to *groups*. Group this note applies to.

usr (varchar(8))

Reference to *users*. User who posted this note.

content (text)

Note contents.

title (int4)

Reference to *strings*. Note title.

date (datetime)

Date and time note was posted.

thread (int4[])

Sequence numbers of all notes above this note in this thread, including the note itself. For instance, if this note has sequence number 456, and this note is in reply to a note with sequence number 123, which began the thread, then *thread* will have value {123,456}.

thread has a unique value within this table.

seq (int4) (primary key)

Sequence number obtained from global sequence generator *seq*.

2.8 Table: *resources*

This table records the name of each shared resource defined by the users.

name (text)

Name of the resource as supplied by the users.

seq (int4) (primary key)

Sequence number obtained from global sequence generator *seq*.

2.9 Table: *strings*

This table stores strings, especially filenames, message titles, and calendar event titles. These are used by reference to sequence number in other tables to reduce redundancy and speed queries.

string (text)

String value. Unique within this table.

seq (int4) (primary key)

Sequence number obtained from global sequence generator *seq*.

grp (varchar(8))

If *string* represents a filename and the filename is in the shared directory space of a workgroup, the name of that workgroup. Otherwise, this field must be null or the empty string.

2.10 Table: *sync_mail*

This table tracks which articles in the *notes* table have been mailed to users who have *emailbody* set to 2. It has a single column named *seq*, of type int4, and a single row whose value indicates the minimum sequence number at which to start mailing articles.

2.11 Table: *uselog*

This table tracks users' usage of the TeamSCOPE system. Each time a TeamSCOPE page is loaded by a logged-in user, a row is added to the table.

usr (varchar(8))

Reference to *users*. Username.

t (datetime)

Date and time the TeamSCOPE page was loaded.

loc (text) The screen, or location, inside TeamSCOPE that was loaded. See function `dispatch_screen` in 'src/main.c' for a list of possible locations.

grp (varchar(8))

Reference to *groups*. Group that user was logged in as.

query (varchar(8))

Complete query string submitted (GET-type queries only).

2.12 Table: *user_group*

This table tracks the membership of users in groups. It also stores the activities last shown to users through web and email interfaces.

grp (varchar(8))

Reference to *groups*. Group name.

usr (varchar(8))

Reference to *users*. Username.

last_read_web (int4)

Reference to *activities*. Last activity shown to the user through the web interface, as confirmed by the user by clicking the **Reset** button.

last_read_email (int4)

Reference to *activities*. Last activity shown to the user through the email interface.

seq (int4) (primary key)

Sequence number obtained from global sequence generator *seq*.

sent_email (bool)

When a new user is added to TeamSCOPE, he is normally sent an email message explaining how the system works. However, this message can be sent at a different time from account creation, so this flag is used to record whether an email has been sent.

may_admin (bool)

True if *usr* is allowed to act as a group administrator in *grp*.

In this table, (*grp*, *usr*) form a unique tuple.

2.13 Table: *users*

This table stores a list of usernames and their associated real names, as well as some other per-user settings.

usr (varchar(8)) (primary key)

Username.

name (text)

Full real-life name of user.

email (text)

Upstream email address of user, if any.

recordable (bool)

If this copy of TeamSCOPE has email recording enabled (for research purposes), a false value in this field will disable email recording for this user only.

idle (int4) Stores the user-configurable number of seconds that the user's session may be idle (no page loads) before the user is required to re-authenticate.

last_usage (datetime)

Date and time of the last TeamSCOPE page load by this user.

emailbody (int4)

Preference of *usr* for being mailed bodies of message board articles. Set to 0 to disable, 1 to enable, 2 to be mailed individual messages as they are posted rather than digests.

primarygrp (varchar(8))

User's default login group.

welcome (bool)

True if user should be shown welcome screen at login time.

2.14 Table: *version*

This table stores exactly one record that specifies the version of the SCOPE database. This table is used by ‘*scope-db*’ to decide what actions are necessary to upgrade the TeamSCOPE database from one version to another.

This manual documents version 3.0 of the TeamSCOPE database format.

major (int4)

Major version number.

minor (int4)

Minor version number.

2.15 Sequence generator: *seq*

This sequence generator is used to generate a unique sequence for each record in almost every table in the TeamSCOPE database, as described above.

3 Filesystem Layout

The table below lists all the files and directories used by TeamSCOPE. Not listed are non-essential files, for instance documentation. The directories listed are those used under Debian GNU/Linux; you may be using a variant directory structure.

<code>‘/etc/cron.d/teamscope’</code>	Schedules TeamSCOPE daily maintenance tasks.
<code>‘/etc/init.d/teamscope’</code>	Used to start and stop the TeamSCOPE chat server.
<code>‘/etc/scope/index.shtml.default’</code>	Skeleton file copied into new teams’ <code>www/index.shtml</code> in their shared folder.
<code>‘/etc/scope/htaccess.default’</code>	Skeleton file copied into new teams’ <code>www/.htaccess</code> in their shared folder.
<code>‘/etc/scope/new-user-mail’</code>	Skeleton file mailed to new users for introductory purposes.
<code>‘/etc/scope/conf’</code>	Configuration file edited before compilation and used at runtime by some TeamSCOPE programs.
<code>‘/etc/scope/reload’</code>	Script run by <code>scope-db update</code> to restart Apache and Exim.
<code>‘/etc/scope/shut’</code>	When this file exists, TeamSCOPE is disabled. To re-enable access to TeamSCOPE, delete the file. The file’s contents, if any, are shown to the user as an explanation of the break in service. Please note: Do not edit this file directly. Most settings must not be changed after compilation; to change these settings, you must recompile and reinstall.
<code>‘/etc/scope/passwd’</code>	Password file for web authentication. Format is <code>‘username:password’</code> , where <i>username</i> is a UNIX username and <i>password</i> is an encrypted password in standard “crypt” format.
<code>‘/etc/scope/apache.conf’</code>	Apache configuration file generated by <code>‘scope-db’</code> . This should be included in the main Apache configuration with an include-file directive.
<code>‘/usr/sbin/scope-chat’</code>	TeamSCOPE chat server.
<code>‘/usr/sbin/scope-db’</code>	TeamSCOPE database configuration and maintenance tool.
<code>‘/usr/lib/scope/’</code>	Directory containing TeamSCOPE programs meant to be used by other programs rather than executed directly.

`‘/usr/lib/scope/www/’`
TeamSCOPE cgi-bin directory. Server alias `‘/scope’` should point to this directory.

`‘/usr/lib/scope/www/scope.cgi’`
TeamSCOPE cgi-bin executable. Must be setuid root and executable by the group that the webserver runs as, but not executable by other users (mode 4774).

`‘/usr/lib/scope/www/ScopeChat.zip’`
TeamSCOPE Java chat client.

`‘/usr/lib/scope/mod_auth_scope.so’`
Apache module implementing TeamSCOPE authentication.

`‘/usr/lib/scope/_int4.o’`
PostgreSQL extension implementing various operations for arrays of integers.

`‘/usr/lib/scope/text.o’`
PostgreSQL extension implementing various operations for text strings.

`‘/usr/share/scope/timezones’`
Generated by `‘scope-db’`. Lists time zones recognized by the local system.

`‘/usr/share/scope/help/’`
Help files generated from `‘scope.texi’`, in mangled HTML format.

`‘/usr/share/scope/www/’`
Directory containing architecture-invariant parts of the webpage.

`‘/usr/share/scope/www/f/’`
Directory containing icons used by the file manager.

`‘/usr/share/scope/www/pb/’`
Directory containing “powered-by icons” for advertising purposes.

`‘/usr/share/scope/www/t/’`
Directory containing TeamSCOPE title graphic.

`‘/var/lib/scope/trusted/’`
Directory that stores one file per active TeamSCOPE session. This is “trusted data,” read and written as root. Non-root users must not be able to scan this directory or read or write files in it. Permissions 0700 are recommended.

`‘/var/lib/scope/untrusted/’`
Directory that stores one file per active TeamSCOPE session. This is “untrusted data,” read and written after root privileges have been dropped. Non-root users must not be able to see the list of files in this directory. This directory should have permissions 1733.

4 Source Code Overview

The source code to TeamSCOPE is written in GNU style published by the Free Software Foundation. See section “Writing C” in *The GNU Coding Standards*, for information on GNU coding style.

TeamSCOPE uses GNU Autoconf to configure itself for the local system. See section “Introduction” in *The GNU Autoconf Manual*, for information on Autoconf.

TeamSCOPE also uses GNU Automake to generate Makefiles. See section “Introduction” in *The GNU Automake Manual*, for information on Automake.

The TeamSCOPE source code is ANSI C89 compliant where possible. Where this is not possible, it attempts to be POSIX compliant. In places where system dependencies are unavoidable, the amount of dependent code has been minimized.

As an introduction to the source code, each source file is given a short description in one of the sections below.

4.1 Screen Managers

The files below each implement one or more screens of the TeamSCOPE user interface. They contain a great deal of HTML input-output. They tend to interact with the web library and support library modules a great deal. There is little interaction among these modules.

‘activity.c’
 ‘activity-cgi.c’
 ‘activity.h’
 ‘activityP.h’

“Recent Activities” and activity search functionality. Some parts are useful to ‘scope-db’; the ‘scope.cgi’-specific parts are in ‘activity-cgi.c’.

‘admin.c’
 ‘admin.h’

TeamSCOPE administration screen.

‘calendar.c’
 ‘calendar.h’

Calendar screen.

‘files.c’
 ‘files.h’

File manager. Interacts with message-board module to allow comments on files.

‘help.c’
 ‘help.h’

Online help screen handler.

‘info.c’
 ‘info.h’

Provides users information on patterns of TeamSCOPE usage by teammates.

`'messages.c'`
`'messages.h'`
 Message board. Implements both the global message board and per-file message boards in conjunction with file manager.

`'options.c'`
`'options.h'`
 Options screen.

`'resource.c'`
`'resource.h'`
 Resource reservation screen. Used by calendar, but separate.

`'search.c'`
`'search.h'`
 File search screen.

4.2 Web Library

These modules perform functions useful in the context of an HTML-generating CGI-bin web-based program.

`'auth.c'`
`'auth.h'`
 Performs user authentication.

`'cgi.c'`
`'cgi.h'`
 Parses CGI parameters in both URL-style and form-data MIME formats in both GET and POST submission styles. Also supports an interactive input style for debugging. Modeled after `'CGI.pm'` from Perl 5.

`'cookie.c'`
`'cookie.h'`
 Parses cookies passed to TeamSCOPE by the web server according to the CGI specification.

`'html.c'`
`'html-cgi.c'`
`'html.h'`
 Two major function families:

- `html_*`() HTML output helper functions. Especially useful for forms and lists.
- `url_*`() URL construction functions. Especially helpful with complex query strings.

These functions are used throughout the screen manager modules. Parts are useful to `scope-db`; the rest are in `'html-cgi.c'`.

`'priv.c'`
`'priv.h'`

Handles dropping root privileges.

`'session.c'`
`'session.h'`

Maintains the illusion of a persistent browser session, in conjunction with the cookie manager, by issuing a session cookie to the client.

4.3 Support Library

These modules provide useful unspecialized services.

`'alloc.c'`
`'alloc.h'`

Memory allocation functions, copyright by Ben Pfaff pfaffben@msu.edu.

`'avl.c'`
`'avl.h'`

AVL tree manager code, written by Ben Pfaff blp@gnu.org, copyright by Free Software Foundation. See section “Introduction” in *The GNU libavl Manual*. GNU libavl is also available separately from the GNU ftp site (<ftp://ftp.gnu.org/pub/gnu/avl>).

`'datetime.c'`
`'datetime.h'`

Date and time parsing code.

`'db.c'`
`'db.h'`

Database interface.

`'error.c'`
`'error.h'`

Displays error messages.

`'id-list.c'`
`'id-list.h'`

ADT for lists of alphanumeric IDs, intended for use with user IDs.

`'identity.c'`
`'identity.h'`

Dumping ground for information about the current and other users; i.e., username, workgroup, uid, gid,

`'pool.c'`
`'pool.h'`

Provides pool-based memory allocation layered on top of `malloc()` and family. Underused in the current TeamSCOPE code. Copyright by Free Software Foundation, written by Ben Pfaff blp@gnu.org.

`'std.c'`

`'std.h'`

String dictionary abstract data type. String dictionaries are used for storing all types of session data. Implemented using AVL trees.

`'str.c'`

`'str.h'`

Two important categories of function:

string functions

Miscellaneous functions for manipulating C-style null-terminated strings.

`st_*`() String abstract data type. Heavily used by screen manager module to construct database queries, etc., and by HTML module to construct URLs.

Contains some code copyright by Free Software Foundation.

4.4 Configuration Files

These source files can be modified at compilation time to change runtime behavior.

`'pref.h'` Configures miscellaneous behaviors that change between OS type.

`'paths.h'` Sets paths and files used by TeamSCOPE at runtime. This file should eventually be eliminated in favor of automatic detection at configuration time.

4.5 `scope.cgi`

This is the main source file for `scope.cgi`. It initializes each of the important libraries, passes execution to the appropriate screen manager, then closes down and exits. `scope.cgi` links with the object files from all the source files described above.

`'scope.cgi.c'`

Contains `main()` function and screen dispatcher for `scope.cgi`.

4.6 `scope-db`

This is the code for `scope-db`, the TeamSCOPE command-line administration tool. It uses several of the files above, but not all of them.

`'scope-db.c'`

`'scope-db.h'`

Contains `main()` function and much of the other code for `scope-db`.

`'mail.c'`

`'mail.h'`

Functions related to mail sending and receiving.

Source Index

A

activity-cgi.c	13
activity.c	13
activity.h	13
activityP.h	13
admin.c	13
admin.h	13
alloc.c	15
alloc.h	15
auth.c	14
auth.h	14
avl.c	15
avl.h	15

C

calendar.c	13
calendar.h	13
cgi.c	14
cgi.h	14
cookie.c	14
cookie.h	14

D

datetime.c	15
datetime.h	15
db.c	15
db.h	15

E

error.c	15
error.h	15

F

files.c	13
files.h	13

H

help.c	13
help.h	13
html-cgi.c	14
html.c	14
html.h	14

I

id-list.c	15
id-list.h	15
identity.c	15
identity.h	15
info.c	13
info.h	13

M

mail.c	16
mail.h	16
messages.c	14
messages.h	14

O

options.c	14
options.h	14

P

paths.h	16
pool.c	15
pool.h	15
pref.h	16
priv.c	15
priv.h	15

R

resource.c	14
resource.h	14

S

scope-db.c	16
scope-db.h	16
scope.cgi.c	16
search.c	14
search.h	14
session.c	15
session.h	15
std.c	16
std.h	16
str.c	16
str.h	16

Database Schema Index

A

AC_CAT_EVENT	3
AC_CAT_FILE	3
AC_CAT_MSG_BOARD	3
AC_FILE_CHMOD (enumeral)	4
AC_FILE_COPY (enumeral)	3
AC_FILE_DELETE (enumeral)	3
AC_FILE_DOWNLOAD (enumeral)	3
AC_FILE_DOWNLOAD_ARCHIVE (enumeral)	3
AC_FILE_MKDIR (enumeral)	4
AC_FILE_MOVE (enumeral)	4
AC_FILE_POST (enumeral)	4
AC_FILE_POST_DELETE (enumeral)	4
AC_FILE_POST_MAILED (enumeral)	4
AC_FILE_POST_READ (enumeral)	4
AC_FILE_RENAME (enumeral)	4
AC_FILE_RMDIR (enumeral)	4
AC_FILE_UPLOAD (enumeral)	3
AC_MSG_DELETE (enumeral)	4
AC_MSG_MAILED (enumeral)	4
AC_MSG_POST (enumeral)	4
AC_MSG_READ (enumeral)	4
activities (table)	3

C

cat (field)	3
content (field)	7

D

date (field)	3, 7
--------------------	------

E

email (field)	9
emailbody (field)	9
event (field)	6

F

file (field)	5, 7
finish (field)	6

G

grp (field)	3, 5, 6, 7, 8
-------------------	---------------

I

idle (field)	9
--------------------	---

L

last_read (field)	5
last_read_email (field)	9
last_read_web (field)	9
last_usage (field)	9

M

major (field)	10
md5 (field)	7
minor (field)	10

N

name (field)	6, 7, 9
note (field)	6

O

owner (field)	6
---------------------	---

P

primarygrp (field)	9
--------------------------	---

R

recordable (field)	9
ref.seq (field)	5
reported (field)	5
resource (field)	6

S

s1 (field)	3
s2 (field)	4
seq (field)	5, 6, 7, 8, 9
seq1 (field)	5
seq2 (field)	5
sharedir (field)	7
source (field)	6
start (field)	6
string (field)	8

T

thread (field) 7
title (field) 6, 7
type (field) 3

U

usr (field) 3, 5, 7, 8, 9

W

webdir (field) 7
welcome (field) 9

Concept Index

A

activities	3
ANSI C	13
Apache	1
Autoconf	13
Automake	13

C

C89	13
calendar event	8
calendar events	6
changes	3
comments	7
conflicts	5

D

database server	1
-----------------------	---

E

Exim	1
------------	---

F

file activities	3
filenames	8
files	11
ftp server	1
full names	9

G

GIF	2
GNU coding style	13
groups	6, 8
<i>grp</i> (field)	8

H

HTML	13
------------	----

L

libgd	2
<i>loc</i> (field)	8
log, usage	8

M

mail server	1
Makefile	13
membership	8
message board	7, 8
message board activities	4

N

names, full	9
notes	7

P

POSIX	13
PostgreSQL	1, 12
posts	7
ProFTPD	1

Q

<i>query</i> (field)	8
----------------------------	---

R

real names	9
resource reservation	5, 6, 7
run-length encoding	2

S

'scope-chat'	1
'scope-db'	1
'scope.cgi'	1
'ScopeChat'	1
screen manager	13
sequence generator	10
shared resource	7
strings	8
synchronous mail table	8
system dependency	13

T

teams 6, 8
trusted data 12

U

unique id 10
usage log 8
usernames 9
usr (field) 8

V

versioning 10

W

web server 1
when (field) 8
workgroups 6, 8

Table of Contents

1	Introduction	1
1.1	Architecture	1
2	Database Schema	3
2.1	Table: <i>activities</i>	3
2.2	Table: <i>catchup</i>	5
2.3	Table: <i>conflicts</i>	5
2.4	Table: <i>event_resource</i>	6
2.5	Table: <i>events</i>	6
2.6	Table: <i>groups</i>	6
2.7	Table: <i>notes</i>	7
2.8	Table: <i>resources</i>	7
2.9	Table: <i>strings</i>	8
2.10	Table: <i>sync_mail</i>	8
2.11	Table: <i>uselog</i>	8
2.12	Table: <i>user_group</i>	8
2.13	Table: <i>users</i>	9
2.14	Table: <i>version</i>	10
2.15	Sequence generator: <i>seq</i>	10
3	Filesystem Layout	11
4	Source Code Overview	13
4.1	Screen Managers	13
4.2	Web Library	14
4.3	Support Library	15
4.4	Configuration Files	16
4.5	<i>scope.cgi</i>	16
4.6	<i>scope-db</i>	16
	Source Index	17
	Database Schema Index	18
	Concept Index	20