Introduction

EventTools is a small online package to provide web-based access to tour and clinic information for NMRA regional and national conventions. It was originally created for the X2011West NMRA convention, and continues to evolve.

EventTools stores a single copy of information, including:

- Clinics
- General/Prototype Tours
- Layouts and Layout Tours
- Miscellaneous Events
- Operating Sessions and requests

From this, EventTools can create order forms, on-line shopping cart, calendars, schedules, printed masters, etc, with completely consistent information. As events sell out, get cancelled or modified, attendees can always be given the most up-to-date and consistent information.

This document is for a webmaster or programmer who is maintaining or adapting EventTools. For user level information, see the Event Tools User Guide, a separate document.

We strongly recommend providing your contention with two instances of EventTools. The first is a "test" one where the information is transient, and people can experiment with the tools without risk of damaging any real data.

The second is the "production" copy contains the real data that will be used for the Convention itself.

The main page:

X2011West Event Tools

Sample page for access to the X2011West tour and clinic tools.

	General Tours	Layout Tours	Other Events	Layouts	Clincs
Review Content	List	List		List	
Enter/Change Content	Enter/Change	Enter/Change	Enter/Change	Enter/Change	Enter/Change
Other Edits		Add Layouts To Tour	Add or remove tags	Quick Entry	Add or remove tags
Index Example	Index	Index	Index	Index	Index
Format Example	Formatted View	Formatted View	Formatted View	Formatted View	Formatted View
X2011west Test Pages (testeventtools only)	X2011west Page	X2011west Page		X2011west Page	X2011west Page
Other Displays	Interactive General/Proto Tour Calendar	Tour Maps (put tour number in URL) Interactive Layout Tour Calendar	Misc Events Grouped by Location Interactive Misc Events Calendar		Clinics Grouped by Location Interactive Clinics Calendar

General Displays	Index Advanced Section Tours All Advanced Section Tours Download Events Calendar Full Interactive Calendar

Technical Information for Webmasters

Briefly: Getting EventTools. PHP structure. HTML structure. Use of CSS.

Getting EventTools

EventTools is available from an open-source project on SourceForge.net at http://eventtools.sf.net and is provided subject to the GNU Public License (GPL). EventTools was written and is maintained by Bob Jacobsen, please contact him directly if you want more information on using EventTools.

To download a copy of the current version use SVN:

```
svn checkout svn://svn.code.sf.net/p/eventtools/code/trunk eventtools
```

We distribute it via SVN so that it's easy to apply patches as the software evolves.

Installing EventTools

This section is a short checklist of operations to install and configure EventTools for basic operation.

- First, you need working PHP and MySQL. PhpMyAdmin will be used for some examples here, but isn't really required.
- Get a copy of the code (see above) and place it somewhere in a web-server file tree so that it can be accessed as, for example, http://myhost/eventtools It is strongly recommended that you put it just under the root level on the server, so that /eventtools URLs refer to it.
- At this point, check that you can see the EventTools index page at http://myhost/eventtools/index.php
- Create a MySql database to store your data or pick an existing one. EventTools puts a common prefix on all its tables within that database, so it can coexist to some extent on a shared database, but a dedicated one is probably better.
- Edit the checked-out access.php file to have your database access information:

  ```
  $opts['hn'] = 'myhost';  // MySQL host name
  $opts['un'] = 'mySqlName';       // MySQL user name
  $opts['pw'] = 'lousyPassword1';          // MySQL password
  $opts['db'] = 'myDB';              // database name to reference
  $event_tools_db_prefix = 'event_2012_';     // prefix on all table names
  ```

- If you have installed Zen Cart and want it integrated with EventTools, make a temporary copy of the define_db.sql, define_views.sql and load_defaults.sql files. If you have not installed and won't be using Zen Cart, make temporary copies of the define_zen_repl.sql, define_db.sql, define_views.sql and load_defaults.sql files. Do a global search & replace in each of those to replace 'prefix_' with the table prefix you selected above, e.g. 'event_2012_'. Have MySQL run all of those files in order.
- At this point, you should be able to display a blank list of layouts without any error messages via http://myhost/foo/eventtools/edit_layouts_all.php

EventTools itself is now up. The next steps are to customize it by editing options in the access.php file, and then to integrate it into your own web pages.

Web pages

EventTools provides complete a complete web interface to enter and administer your event data. The entry point for this is the index.php page in the EventTools directory.

- Pages with names like edit*.php are the phpMyEdit add/change/delete pages.
- Pages with names like list*.php are comprehensive dumps of content, useful for checking.

EventTools also provides access to the data for inclusion in your own formatted pages, e.g. for display to the user. For examples of how that's done (which may also be useful when checking your data entry):

- Pages with names like format* display various forms of formatted output tables, with only minimal actual formatting. These serve as examples of how to get the data for your own pages.
- Pages with names like index*.php are minimal-size indexes, suitable for finding a specific entry in the longer formatted pages.

It's not intended that any of those pages will be available to convention attendees directly. Rather, the index* and format* pages will provide examples for creating main-site pages.

URL Queries

URL queries are used to select entries, and control the ordering of the EventTools output. For example, appending "?presenter=Bill" to an EventTools URL will, by default, display only clinics with "Bill" as part of the presenter name. Appending "?order=date" will order the display by start date and time. See the "samples.php" file for examples, and the "parser.php" file for structure and a list of available query terms.

Select Term	Clinics	Misc Event	General Tour	Layout Tour	Layouts
changed For user at the convention, indicates the entry changed after close-out date	Y	Y	Y	Y	Y
tag Indicates that a tag is attached to this event or clinic, e.g. tag=DCC	Y	Y			
date Day on which this occurs, e.g. ?date=07-12	Y	Y	Y	Y	Y
number	Y	Y	Y	Y	Y
id	Y	Y	Y	Y	Y
layoutid Allows you to reference the ID of the layout, without confusion with the ID of a layout tour				Y	Y
name Matches any string in the name	Y	Y	Y	Y	Y
layoutname Matches any substring in the layout name				Y	Y
owner Matches any substring in the first or last name of layout owner					Y
access Matches any substring in the accessibility				Y	Y

Select Term	Clinics	Misc Event	General Tour	Layout Tour	Layouts
code					
presenter Matches any substring in the first or last name of clinic presenter	Y				
scale Matches any of one or more scale strings, e.g. scale=HO,N Matching is text matching, so ?scale=N matches Nn3				Y	Y
gauge Matches any of one or more gauge strings, e.g. gauge=HO,N Matching is text matching, so ?gauge=n3 matches n35				Y	Y
prototype				Y	Y
layout Matches in layout name				Y	Y
match Generic text search through (most) fields	Y	Y	Y	Y	Y
status Greater than or equal check on status field	Y	Y	Y	Y	Y
type Used to check for leading G or P in the tour type, hence general or prototype tour			Y		
where Allows providing a MySQL “where” clause for inclusion in the query	Y	Y	Y	Y	Y
control Layout control system				Y	Y
era					Y
class					Y
theme					Y
scenery					Y
size					Y

The presentation sequence of the results can be controlled with an “order” term in the URL query.

These include:

Ordering Term	Database Field(s)
?order=status	status_code
?order=lstatus	layout_status_code
?order=name	name
?order=number	number
?order=date	start_date
?order=time	start_date
?order=price	tour_price
?order=presenter	clinic_presenter
?order=clinic_room	clinic_location_code
?order=misc_room	misc_location_code
?order=city	layout_city
?order=owner	layout_owner_firstname, layout_owner_lastname
?order=lastname	layout_owner_lastname
?order=layoutname	layout_name

HTML tables

The formatted output of EventTools is HTML tables. Each element (table, tr, td and span for content) is given a specific class name so that it can be located by JavaScript or CSS.

CSS

All formatting is done via CSS. EventTools itself does only minimal formatting. Instead, EventTools puts each HTML element into a named class, so that CSS can directly format it.

For more information, see the separate "EventTools and CSS" document.

Updates

EventTools is maintained in SVN. This makes it easy to update the code and documentation so that it stays current with the most recent version. Doing

```
svn update
```

will update the code with any changes since the last update, and give you a list of changed files. Your own local changes will be preserved.

Sometimes, the code requires changes to the MySQL table layout, such as different sizes for columns or new columns or tables. The "update_table.sql" file is maintained to handle these. Please read its comments after each update, and execute the file in your MySQL server if needed. Remember to take

backups, and that in general you'll have to update the "prefix" strings to be what you're using on your local installation. It may also tell you to run the "define_views.sql" file afterwards, in which case you should do that.

Email setup

Several pages can email information to attendees and hosts. The configuration of this can be tricky, as services like AOL and Yahoo are not particularly interested in trusting email from your site.

There are several settings in access.php that control this.

`$event_tools_registrar_email_address` – this is the default value for "from:" and "reply-to:" fields in sent emails
`$event_tools_registrar_email_prefix` – this is provided as a prefix on the subject line to make it easier to filter out EventTools emails in your incoming mailbox

AOL (and to some extent Yahoo) are likely to reject email that they think don't come from the sending domain. E.g. if your email address is @gmail.com, but gmail.com doesn't resolve to the IP address you're sending from, AOL won't accept your mail. The solution is to create an email address on your host machine and put that in the "From:" field, and put the registrar's real email in the "Reply-to" field.

Security

Only minimal PHP header access security is provided via the security.php file and calls. Most format* pages are unrestricted. The list* pages, which may contain private contact information, have access restricted to a class of users. The edit* pages, which allow read and write to all information, are restricted to a (smaller) class of users.

Security is provided by the secure.php file, invoked at the top of pages to be secured. It in turn uses WWW-Authenticate and the $_SERVER['PHP_AUTH_USER'] mechanism to identify the user name, which it checks against the (prefix)_eventtools_users table in MySQL.

The access.php tools has two options that control that default secure.php file:

- event_tools_require_user_id – the user has to provide a user ID (email address) for logging purposes, but it's not verified
- event_tools_require_user_authenticate – the user has to provide a user ID (email address) and password which is checked against the (prefix)_eventtools_users table; access is only granted if there is a matching row

EventTools itself does not provide an edit page for the (prefix)_eventtools_users table. Maybe someday, but for now, use admin-level tools like phpMyEdit to add and modify entries in that table.

Technical Information on EventTools Structure

This section discusses the internal structure of EventTools. It will help if you have to install or modify it.

EventTools depends on several other open-source products:

- MySQL (part of the usual LAMP stack on a web server) – the database access in EventTools is coded in PHP for mysql. No attempt has been made to write database-independent code.

- PHP – At least version 5.3 is required (large parts of EventTools might work with older versions, but EventTools uses the calendar handling in 5.3 when working with schedules)
- phpMyEdit (http://www.phpmyedit.org/) is used to create the web forms to add/edit database data.
- SVN – The master development copy of EventTools is kept in an SVN repository. If you want a copy of EventTools, you obtain it as SVN checkout so that you can easily merge updates in the future.

PHP file structure

All PHP files are individually included by the top-level PHP web page, so that any needed directory structure can be added as part of creating the web pages.

The "access.php" page must be included first in every case. It provides all necessary PHP customization, including the MYSQL access data, database names, and various global options. It should be the only file you need to change to move EventTools from one location to another.

MySQL Use

EventTools prefixes each of its MySQL table names with a specific prefix made of two parts. One is specified in the access.php file, so that you can have multiple instances of EventTools ("test_", "prod_" in a single database. The second is "eventtools_" to separate the EventTools tables from others. A typical table name is "test_eventtools_clinics".

"Events" are held in four tables:

- _clinics
- _layout_tours
- _general_tours
- _misc_event_tours

The first few elements in each row are common, which allows common processing:

- id – a small int, used for e.g. links. Unique, internal primary key
- number – small string (e.g. "L123"; yes, we know that's not a number, but that's what people always call it)
- name – also a title for the clinics
- start_date - "2011-07-03 13:00:00" (standard MySQL format)
- end_date – ditto
- description – used in summaries, etc
- status_code – keyed to the _tour_status_values table

Data Items

EventTools either needs to have Zen Cart installed in the same MySQL database, or have it's own copy

of certain customer tables loaded. The "define_zen_repl.sql" file will install these if you're not using Zen Cart.

The "define_db.sql" file will configure the MySQL tables for an EventTools instance. That needs to be followed by "define_views.sql" to create the cross-table views. (You should also run "define_views.sql" after any updates to table structure so that the views pick them up the changes)

Note that define_db.sql has to give the complete table names, including the prefix, you should do a bulk edit from "test_eventtools_" if you want to use another prefix, then update the access.php file.

The "load_defaults.sql" file will load defaults for clinic location codes, handicapped access codes, and event status codes. You can either execute it, or enter those manually.

The "load_db.sql" file will load a preliminary set of content for testing purposes. You shouldn't do this on your production database.

Zen Cart Interface

X2011west used Zen Cart for its online store. This section describes the interface between EventTools and Zen Cart.

We decided early on not to mix database tables between Zen Cart and Event Tools. Although their tables live in the same MySQL database, we're accessing them separately with the two separate applications.

Loading Zen Cart from EventTools

Category matching e.g. Layout Tours as category

Conversion routine using Easy Populate:

Use of event status codes, add-to-cart links:

Accessing Zen Cart for Attendee (Customer) Contact Information

The OPSIG organizers wanted to cross-check operating session requests with attendee name and address. This was done by a read-only view into the Zen Cart tables from the EventTools tables. If you are not using Zen Cart, the define_zen_repl.sql file will define substitute tables to hold attendee information such as name, address, etc. Alternately, if you have that information in an other table, the view can be redefined to access it from there.

Special Cases & Lessons Learned

The X2011west organization encountered a number of special cases. Many of them were built into EventTools, and become not-so-special. This section describes how the remaining cases were handled manually.

- NASG Banquet, NASG Welcome Get Together, X2011west Nonquet: These are scheduled items, for which tickets are sold. That makes them like general tours. At the same time, the organizers wanted them to appear in the general schedule, which makes them Misc Events.

- Jelly-Belly tours: These are purchased general tours, but the ticket holder is allowed to get zero-cost tickets to bring along their children.
- Modeling with the Masters: Formally clinics, these require advance purchase, with only certain scales available for certain clinics.

Missing data handling

Entry forms, esp the email form for layout owners.

Use of prod & test databases

Zero price (default is -1 for "no data) and self-scheduled (no times, soft times) tours (2011-0707 00:00:00 is a blank time)

Configuring the access.php File

The access.php file is the central configuration file for EventTools. It contains the necessary information to access the underlying MySQL database. It also contains the variable definitions that control presentation and selection of EventTools data. The master copy from SVN is heavily commented, and we recommend that you work through it line-by-line when you first install EventTools. A few specific parts are described more in this section.

Note that because access.php contains access control information, it is very important that it never be publicly available. In particular, don't commit it to your local SVN repository!

Components

Eventtools supports a number of different kind of events: Tours (of several types), clinics, general events, and operating sessions. If your convention doesn't have all of those, you can suppress them with options in the access.php file.

Free-form vs Constrained Entry

A number of fields can either have free-form contents, or have only specific allowed variables. You select this in the "constraints" section of the EventTools access.php file at the level of individual variables. For example, setting:

$event_tools_constrain_scale = FALSE;

allows you to enter arbitrary text in the "Scale" field of the layout entries. "HO", "N and N30", "unknown", and "1:87" are all acceptable values. This allows a lot of flexibility, but makes it harder to create powerful search and selection capabilities.

On the other hand, setting:

$event_tools_constrain_scale = TRUE;

configures EventTools to only allow specific values in the "Scale" field. A constraint table contains the acceptable values, and an enter/edit page is provided to make it easy to define convention-specific

values.

Note that the value is directly stored in the MySQL row, regardless of whether the field is free-form or constrained. This means that changing a constraint row doesn't change any of the places where the value has already been entered; they'll still have the old text.

Access and Authentication

The EventTools database tables can only be modified via the edit_* files, i.e. the myPhpEdit pages. Some sensitive information (layout owner phone and email, etc) can also be seen through the list_* pages, and some of the administrative tools (the mapping tools, etc).

All myPhpEdit modifications are logged to a “changelog” table in the database automatically by myPhpEdit. If the access.php file includes $eventtools_require_user = TRUE, the user will be prompted on first access to a page for their user ID to be put in that logging.

(Email copies for audit & backup)

If $eventtools_require_authentification = TRUE, as simple table-based access check will be done via the users table. X2011west has put the email address of administrators as their user IDs.

This is not a high-security access control mechanism, and it's not intended to be one. For more stringent requirements, it should be possible to replace the mechanism in the secure.php file without changes elsewhere.