Release notes for ShakeMap r1208:

Sorry for the lengthy install notes, but the new config file management takes a little explaining. There are also two special notes for U.S. regional networks.

To perform this update, you'll need to install the Perl module DateTime. It is available through CPAN and many of the package managers that are used by various flavors of Linux/Unix (yum, dpkg, MacPorts, etc.). **Solaris users**: I've had reports that installing DateTime required a fairly significant effort. DateTime has quite a few dependencies, so use of CPAN or a package manager is strongly recommended. See the Appendix for a list of DateTime's dependencies.

You will also need the system command *merge* (i.e., the three-way file merge). Most systems have this already installed as part of the RCS or CVS packages, but for **Solaris**, in particular, you may need to install one or the other package (RCS or CVS) to get *merge*. See **Note to Solaris Users**, below.

As usual, "svn –r1208 update" and "make" do the update, but before you do anything please read the section on the new config file update program "configconfig2," below. If you have an unofficial version of ShakeMap with a release number greater than about 740 (the last official release was 687), then you already have configconfig2, but it's still worth reading below to know what it is doing.

There is a new config file update program **configconfig2**. The previous method of updating configuration files was not optimal. This revision includes a new approach based on the 'merge' system command. Merge does a three-way compare to combine separate changes to an original. It can generally do this without user intervention unless there are conflicts between the changed versions. It should greatly simplify the process of upgrading ShakeMap when config files change, especially after this initial installation.

Check that your system has 'merge' the three-way file merge command. Every system I've encountered has it by default, but if yours doesn't, you'll need to install it. It's typically part of CVS or RCS or other version control systems.

From an upgrade point of view, configconfig2 works much like its predecessor: it reports which files have been added, which have been removed, which have been modified (creating a backup copy of your old config file with a .BAK extension), but now also reports which updated files require action on your part. The new version will also operate on the files in your "zone" (or "zone2") directory, and can update event-specific configs in your data directory.

Because this is the first time we've used it, I recommend a little extra caution. Before you do 'make' for this upgrade:

1. Edit existing config files and remove unwanted comments of the type:

   # <<<< Local Modifications >>>>
   # addon_code   : smgl
   # <<<< End local modifications >>>>

   configconfig2 doesn't delete your comments, so this kind of junk will hang around forever if you don't delete it. Don't forget the files in the zone (or zone2) directory if you use it.

2. The config directory seems to accumulate a lot of cruft on many systems. Now might be a good time to clean it up. In particular, may wish to get rid of old, unnecessary *.BAK files.

3. Copy the config directory for safety:

   % cd <shake_home>
   % mkdir config_safe
   % cp –r config config_safe

   You can delete config_safe once you're sure the config process went cleanly.

4. Proceed with the normal update and make process.

Once you've run 'make,' you should review the output to determine if any files had conflicts. If a file is given an 'ok' it did not have conflicts. Files with conflicts will say so. You **must** edit these files and resolve the conflicts. Conflicts within the file will be indicated with blocks that look like this:

<<<<<<< your_version
[zero or more lines of content from your version of the config file]
=======
[zero or more lines of content from the new version of the config file]
>>>>>>> new_version

Some conflicts will be simple – selecting your version of a parameter setting over the generic one, for example. Other conflicts will involve larger chunks of content. I've noted that conflicts often emerge when I have removed content (i.e., a parameter and its documentation) from a file – in those cases you'll likely be deleting the lines in the "your_version" chunk of the conflict block. When in doubt, refer to the new version of the config file found in <shake_home>/src/cfgsrc and preserve its structure. (Note: there may be more than one block of conflicts in any given file; search until you've resolved them all. Also, make sure you remove the lines demarking the conflict or the ShakeMap programs will choke on them.)

I've done this on a few installations, and it has been relatively painless. One issue that does come up is handling config files with non-standard names (e.g., transfer.conf.primary & transfer.conf.secondary). In the 'config' directory, configconfig2 will only update *.conf files. For example, assume that the machine you are working on is in ShakeMap backup mode, so transfer.conf is the secondary version configured not to send out its products (e.g., it is either identical to, or a symbolic link to, transfer.conf.secondary). It will be updated by configconfig2 when the initial 'make' is done. To finish the update, edit the files as necessary to fix any conflicts that occurred with the merge, then copy transfer.conf back to transfer.conf.secondary. Next, copy transfer.conf.primary to transfer.conf and run configconfig2 again (from the config directory run "../bin/configconfig2 –shake"). Again, edit transfer.conf if necessary to resolve the conflicts, then copy it back to transfer.conf.primary. Finally, copy transfer.conf.secondary back as transfer.conf to leave your system in its initial state. (If you use links instead of copies, it's a similar procedure: you just point the links back and forth rather than copying files.)

By default, the make process will also run configconfig2 on the files in the 'zone' (and 'zone2') directory. In this case, since the files don't end in .conf, the configconfig2 searches for files that start with the base name of any of the *.conf files in <shake_home>/src/cfgsrc, and updates all of those. You can run configconfig2 manually on the zone directories with the "-zone" flag.

Finally, you can run configconfig2 with "-data" and it will **recursively descend all the directories starting from the calling directory** and try to update anything that looks like a ShakeMap config file. While the program has some smarts built in (it won't try to update the config files in ShakeMap's src directory, for instance), it has the potential to do a lot of damage. (Don't call it as super user from '/'.) There is a '-dryrun' flag that will tell you what the program would do without actually doing anything – to be safe, I strongly recommend you always try -dryrun first when using -data. I also recommend that before you do your data directory wholesale, you try the program on a few specific events to get an idea of what it is going to do. Remember: the starting directory will be whatever directory you are in when you execute the command (so don't run it from "<shake_home>/bin".)

If you are using custom config files, Pete Lombard shared a tip:

Some of us have extra programs and config files added to ShakeMap. With the old "configconfig" program, it made some sense to copy these config files into src/cfgsrc. But with the new "configconfig2", this gets too complicated with the extra files on src/cfgsrc/Old/. Its better to remove any copies of the "extra" config files that we may have put into src/cfgsrc. […] (configconfig2 will note these files as old or unrecognized, but will otherwise leave them alone.)

db2xml – new behavior – **Attention AQMS users!**

This update resolves a number of operationally unsatisfying characteristics of db2xml, especially when used with more than one database (real time and archive databases, primary and backup systems, etc.) Most of the hard work here was done by Pete Lombard, so thanks go to him for his efforts.

The major problem was the result of a combination of issues: 1) Because of unknown (and apparently unknowable) database replication times, it is possible that when ShakeMap is triggered, the real time database will have a more complete set of amplitudes than the archive database; but because the archive database gets the "exotic" amplitudes, it will ultimately have the more complete set of amps. There will also be times and circumstances where a union of the two databases' sets of amps will provide a more complete set of amps than either of the two individually. 2) If a database is down or inaccessible, db2xml might a) wait a long time, and/or b) return nothing.

These issues could result in a ShakeMap with few (or no) amps, and possibly delay the map's production. The solution has two parts:

1. Oracle clients can wait a long time before timing out on requests to host computers that are down. Pete writes:

   Google for "oracle tcp timeout" and you get several links. Basically, you create a file sqlnet.ora with a line like:

   tcp.connect_timeout = 5

   This file goes in network/admin/ under your $ORACLE_HOME directory. If the client cannot connect within the connect_timeout (seconds), the client returns with a "connection timed out" error.

   I have tested this feature, by configuring a non-existent host into db.conf. The connection does indeed time out at the specified timeout.

   We recommend you talk with your DBA, if necessary, and set this up for your ShakeMap clients. The timeout should be long enough that it doesn't trigger for the occasionally slow network, but short enough not to delay db2xml for too long.

2. There is a new version of db2xml that will fork off the database queries and kill them if they take too long. Please review the documentation in db2xml.conf for *fork_mode* and *max_wait*. Also note that the default *query_mode* is now 3, which creates a data file for every database that returns data. If you take advantage of the new features (and we recommend you do), please do some timing tests – it will likely be different for different systems/networks/etc. Pete writes:

Regarding the default value for max_wait (120 seconds): that may seem long when testing with moderate earthquakes. But for the M6.9 and aftershocks yesterday, the slowest child was taking about 90 seconds to complete. So I feel comfortable with 120 seconds for max_wait.

Also:

I recommend running db2xml with the -verbose flag, to see the times of transactions with each database. I added this flag to my retrieve.conf file for db2xml.

Auxiliary (a.k.a. "Zoom") Maps

(This note only applies to US regional networks that are using the USGS PDL system to send ShakeMaps to the USGS. However, other ShakeMap operators should be warned that the tilde ("~") character is now a reserved symbol and should not be used in event IDs, except to designate an auxiliary map.)

There have been requests for the ability to have auxiliary ShakeMaps that are associated with the primary ShakeMap for a given earthquake. The USGS web site will soon support this capability, and this ShakeMap release contains a mechanism to designate an event as an auxiliary map. The short explanation is to create a ShakeMap event ID that consists of the earthquake event ID followed by a tilde followed by an identifying string. For example, for and earthquake with the event ID "12345678", the primary ShakeMap would use that same event ID ("12345678"), and an auxiliary ShakeMap might use the event ID "12345678~zoom" (where "zoom" could be any string (within reason): "aux1", "los_angeles", "plus_2_sigma", etc.) This approach has the advantage that it will use a unique ShakeMap data directory, and will appear as a separate event on the ShakeMap-generated web site, but will let the USGS know that the auxiliary maps should be associated with the primary map. The primary map will appear first in the list of available maps, and will be marked with a green checkmark to indicate that it is the preferred map. Auxiliary maps and maps from non-authoritative regions will appear in a list (with thumbnails) below the primary map. See the pdl_code documentation in transfer.conf for details.

The designation for scenarios is that the root event ID ends with "_se" (or you systematically use the –scenario flag when running ShakeMap programs). Ending the extension with "_se" has no effect. The following table summarizes the options:

Event ID	Associated with Event Id	Scenario?	Primary or auxiliary?
12345678	12345678	no	primary
12345678_se	12345678_se	yes	primary
12345678~zoom	12345678	no	aux

12345678_se~zoom	12345678_se	yes	aux
12345678~zoom_se	12345678	NO!	aux
12345678_se~zoom_se	12345678_se	yes	aux

db2xml and **eq2xml** have been modified so that when called with a compound event ID (like "12345678~zoom" they will create a directory structure using that ID, but will query the database(s) with the base event ID (in this example "12345678"). One or two other rarely-used programs in the src/xml directory have had similar modifications – they create the path using the full event ID, but look for a file (in the directory "raw" for example) using the base event ID.

To run an auxiliary map, you would typically make a new input directory for the event (e.g., "<shake_home>/data/12345678~zoom/input"), copy the input files from the base event, edit event.xml to set the correct "evid," and then run *shake* with the –dryrun flag. You can then modify the default commands to your zoom map specifications (e.g., change the "lonspan" in *grind*), and then run them to produce the new map. AQMS users could skip the first steps and just run *retrieve* for the new event ID, and then run *shake* with the –dryrun flag.

Running auxiliary maps automatically for every event requires more extensive modifications (including, possibly, a second ShakeMap installation), and is beyond the scope of this document.

Note: If you are already making auxiliary maps using the same event ID as the primary (but through a separate ShakeMap installation), you can tell the USGS that the map is auxiliary by creating a unique product code in transfer.conf. For example, if your current setting for "pdl_code" is something like "pn<EVENT>", you would continue using that in transfer.conf for the primary events, and use something like "pn<EVENT>_zoom" for the auxiliary map(s) (here, "_zoom" can be any string within reason, and no tilde is necessary). The only problem with this solution is that it does not create a unique event ID for the ShakeMap-generated web site, so only the most recently transferred event with that ID would appear. If you roll your own web site, this shouldn't be an issue and you can handle multiple maps with the same ID however you want.

Other significant changes in this release:

contour – There was a bug in contour that would cause a segmentation fault under certain relatively rare circumstances (large mapped area, high ground motions, choppy Vs30).

retrieve – No longer exits when run on a scenario. If you don't want 'retrieve' to run on scenarios, set "scenario_skip : retrieve" in shake.conf (this is the default setting, but may not be set in your shake.conf for one reason or another). On the other hand, if you do

want retrieve to run for scenarios, make sure to remove "scenario_skip : retrieve" from shake.conf.

mp – I added source code for the Metadata Parser to the distribution. There is no reason to change your current mp, I've added this for convenience of future installs. mp can be made and installed in bin by doing "make mp" in <shake_home>. Remember that you set the path to mp in genex.conf. Note that this compile has not been tested on a wide range of systems yet, so some tweaking may be necessary. Let me know if you try it and it gives you problems, and send me the fixes. I'll try to include fixes in future releases. This version differs from the official USGS version in that it contains only the code to make 'mp' and not the other programs that come with the source download, and I've modified the code as necessary to eliminate a number of warnings from the compiler. Functionally, it should be identical to the official version.

zoneconfig2 – This is not a replacement for zone_config, and it is unlikely to be used by most operators. It is a tool that the Global ShakeMap (GSM) system in Golden, CO, uses to determine GMPE/IPE/GMICE choices for global earthquakes. It has been added to the distribution purely for our convenience when upgrading the GSM systems. It is not installed by default. It is Python code and has its own set of dependencies. The only reason I mention it here is that you may come across it, or the PYTHON macro in include/macros, and wonder what they are. Just ignore both.

New config file: timezone.conf. This config file supports more flexible handling of dates and times in ShakeMap. Please read through the documentation in timezone.conf to understand the new options. In general, you now have considerably more control over the format (and time zone) of times and dates printed on the maps and web pages. In addition, earthquake information (in event.xml) can now be in any time zone (with appropriate setting of the 'timezone' attribute), but we hope no one makes use of this questionable "feature" – event info should really be in GMT/UTC. An unspecified 'timezone' in event.xml defaults to GMT. The new configurations generally **do not apply to the timestamps in the ShakeMap log files and screen output**, which will generally be in the computer's notion of **the local timezone**. As part of this revision, the parameter **use_utc** in web.conf has been **removed**. In addition, dependence on the Perl module **Time::y2038** has been **removed** from the code and is no longer required for ShakeMap installation (actually, it was recommended, not required, but it is still gone). The new features depend on the Perl module **DateTime**, so that **must be installed** on your system to perform this update.

New GMPE module: ASB13 (Akkar, et al., 2013) is a new GMPE for crustal earthquakes in **Europe and the Middle East**.

grind, **shake** – There is a new flag, **-comment**, that allows the operator to add a quoted line of text to info.xml.

grind – Fixed a bug that sometimes caused the MMI bias to be excessively large when most observations were (or mapped to) MMI 1.0.

mapping – There is a new flag, **-nohinges**, that suppresses the plotting of the dotted lines along the hinges of multi-plane faults.

info.xml – The parameters <param>_max (where <param> is one of "mmi," "pga," "pgv," "psa03," etc.) now hold the maximum gridded value on land, and we've added new parameters <param>_max_grid that hold the maximum amplitude anywhere in the grid.

Added **BB.pm module** for **Beyer and Bommer (2006)** corrections to amps and sigmas for converting from various ground motion definitions (e.g., arithmetic mean, random component, geometric mean) to maximum component; updated all the GMPE modules (except those with their own built in corrections, and Kanno06 which uses vector sum, which is not supported). This was formerly handled in an inconsistent, ad hoc way among the GMPEs. Some GMPEs' amplitudes may change somewhat, but the effect should not be large – on the order of a few percent.

To be consistent with the online maps, station and DYFI icons are now triangles and circles, respectively, in the **KML**. The KML in general has been revamped (again). The eventID.kml now only has two links: one to eventID.kmz and one to the organizational logo image configured in genex.conf. The file eventID.kmz is new – it contains all the various pieces, icons, images, etc., in one self-contained file. You can use eventID.kmz without the support of a web server, so it's handy when working on events you're not yet ready to push to the web. All of the earlier KMZ files still exist, though at this point they're primarily for the USGS web team to add to their "event" KML (which includes PAGER, DYFI, and other products).

mapping.conf: Added parameter '**map_cities_on_pgm**' to allow plotting of city names on PGA, PGV, and PSA maps. Also added the same parameter to the programs *plot_vs30* and *view_rcg*.

queue and **queue_qdds**: Improved timestamps of logged output per Bob Dollar and Gary Gann's suggestions.

shake, **cancel**: Change subject of email to "runtime report" rather than "error report" if normal exit status. Also added event ID to subject line. (Changes per Bob Dollar/Gary Gann.)

Makefiles: The make process should now handle errors somewhat better – by quitting more reliably when errors are encountered. Operators should still scan through the output to look for problems, but hopefully these modifications will make major problems more apparent. The Makefiles are also faster now.

Many bug fixes and minor enhancements. Most functions now check earthquake depth, and if it is less than zero, it is set to zero.

Note to Solaris Users

Pete Lombard writes:

If you install the GNU "rcs" package or source code to provide "merge", be sure you also install the GNU "diffutils" package or source code to provide "diff3". The version of diff3 provided by Solaris is NOT compatible with what GNU merge expects. If installing from source, be sure to install diffutils first. Then set the DIFF and DIFF3 environment variables to the locations of your newly installed diff and diff3 programs, respectively. Otherwise "merge" will get built with the wrong path to diff3.

References:

Akkar, S., M. A. Sandikkaya, and J. J. Bommer (2013). Empirical ground-motion models for point- and extended-source crustal earthquake scenarios in Europe and the Middle East, *Bull. Earthquake Eng.*, online 31 May 2013, DOI 10.1007/s10518-013-9461-4.

Beyer, K., and J. J. Bommer (2006). Relationships between median values and between aleatory variables for different definitions of the horizontal component of motion, *Bull. Seism. Soc. Am.*, **96**, 1512-1522.

Appendix – Perl modules required for module DateTime

Pete Lombard kindly supplied a list of modules (in the order they should be installed) required for DateTime to install. Again, I highly recommend using CPAN or a package manager to do the install because they will (usually) take care of all of this for you.

parent
version-0.9906
Test::Simple
CPAN::Meta::Requirements
File::Path
File::Temp
CPAN::Meta::YAML
Parse::CPAN::Meta
CPAN::Meta
Perl::OSType
Path::Tools
Locale::Maketext::Simple
Params::Check
Module::CoreList
Module::Load
Module::Metadata
Module::Load::Conditional

IPC::Cmd
ExtUtils::CBuilder
ExtUtils::Install
ExtUtils::Manifest
File::Copy::Recursive
ExtUtils::Command
JSON::PP::Compat5006
ExtUtils::Manifest
ExtUtils::MakeMaker
ExtUtils::ParseXS
Pod::Escapes
Pod::Simple
podlators
Test::Harness
Module::Build
Try-Tiny-0.22
Test-Fatal-0.013
Test-Tester-0.109
Test-NoWarnings-1.04
Test-Deep-0.112
Module-Runtime-0.014
Test-Requires-0.07
Module-Implementation-0.07
Attribute::Handlers
Params-Validate-1.10
Exporter-Tiny-0.038
List-MoreUtils-0.400_009
DateTime-Locale-0.45
ExtUtils-Config-0.007
ExtUtils-Helpers-0.022
ExtUtils-InstallPaths-0.010
Getopt::Long
Module-Build-Tiny-0.036
Package-Stash-XS-0.28
Dist-CheckConflicts-0.11
Package-Stash-0.36
Params-Util-1.07
Sub-Install-0.927
Data-OptList-0.109
Class-Load-0.21
Class-Singleton-1.4
DateTime-TimeZone-1.69
Test-Warnings-0.014
DateTime-1.10