Utility Scripts

Edit me

A number of valuable Perl scripts are distributed with ShakeCast. This Appendix documents these utilities. For the most part, these scripts are internal to the system and normally users many need only a few of them, but users with complicated databases and needs may benefit at least an awareness of their functionality.

Ground Shaking Estimation Tool

NAME

facility_feature_shaking.pl – Ground Shaking Estimation Tool for Facilities with Complex Geometry

SYNOPSIS

facility_feature_shaking.pl [option ...]

DESCRIPTION

The **facility_feature_shaking.pl** utility is used to generate ground-shaking estimates for facilities with complex geometry. It reads one event id and one version from the command line. Output of the script is saved in facility_feature_shaking.xml under the directory of the specified event.

The default grid processing routine of ShakeCast handles facilities with either point or rectangle shapes. Recognized complex shapes include circle, polyline, and polygon. The shaking information is considered as secondary and cannot be used as notification thresholds.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or by the system workers through dispatcher tasks.

OPERATIONS

event

Specify ID of the event to process.

version

Specify Version of the event to process.

OPTIONS

–verbose

Display more detailed information about the progress of the analysis. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

Probabilistic Fragility Estimation Tool

NAME

facility_fragility_stat.pl – Probabilistic Fragility Estimation Tool for Facilities with Probability Fragility Curves

SYNOPSIS

facility_fragility_stat.pl [option ...]

DESCRIPTION

The **facility_fragility_stat.pl** utility is used to evaluate both CDF and distribution of likelihood of damage states for each identified components of the facility. It reads one event id and one version from the command line. Full fragility curve interpretation is one of the most time consuming processes of ShakeCast. The utility saves results of the analysis in both binary form “frag_prob.hash” and in text

form “frag_prob.json”. The binary data is a fully structured data storage that captures a snapshot of the final data output that can be used by subsequent processes without repeating the same computation. The text-based JSON output is designed for presentations through the web interface.

The shaking information is considered as secondary and cannot be used as notification thresholds. The probability fragility tool provides detailed evaluations to complement the basic fragility and notification processes.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or by the system workers through dispatcher tasks.

OPERATIONS

event

Specify ID of the event to process.

version

Specify Version of the event to process.

OPTIONS

–verbose

Display more detailed information about the progress of the analysis. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

Tool for Evaluating Exceedance of Regulatory Levels (Nuclear Power Plants)

NAME

facility_regulatory_level.pl – Tool for Evaluating Exceedance of Regulatory Levels for Nuclear Power Plants

SYNOPSIS

facility_regulatory_level.pl [option ...]

DESCRIPTION

The **facility_regulatory_level.pl** utility is used to evaluate exceedance of regulatory levels for nuclear power plants. Regulatory levels include SL1/OBE, SL2/SSE, and Reg. 1.166 Appendix A. It reads one event id and one version from the command line. Results of evaluations are saved in the output file “facility_regulatory_level.xml” to be used in preparing the PDF reports and to be displayed through the web interface.

The shaking information is considered as secondary and cannot be used as notification thresholds. Although the script is created for the nuclear industry as part of the project requirement, it can be modified to provide user-specific criteria as a rule-based analysis tool. The regulatory level tool provides detailed evaluations to complement the basic fragility and notification processes.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or by the system workers through dispatcher tasks.

OPERATIONS

event

Specify ID of the event to process.

version

Specify Version of the event to process.

OPTIONS

–verbose

Display more detailed information about the progress of the analysis. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

USGS Earthquake JSON Feed Parser Tool

NAME

gs_json.pl – USGS Earthquake JSON Feed Parser Tool

SYNOPSIS

gs_json.pl

DESCRIPTION

The **gs_json.pl** utility is used to parse the USGS earthquake JSON feed and selected products from the web server. Currently accepted earthquake product types include ShakeMap, DYFI?, PAGER, Earthquake Location Map, Historical Moment Tensor Map, Historical Seismicity Map, Tectonic Summary, Origin. The parser follows the JSON feed of individual earthquake products to download the selected products for use by the local ShakeCast system. Origin, ShakeMap, DYFI? and PAGER will invoke a Dispatcher task to handle the downloaded products.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or queued to be handled by the system workers.

ShakeCast Heartbeat Generator

NAME

heartbeat.pl - ShakeCast Heartbeat Generator

SYNOPSIS

heartbeat.pl

DESCRIPTION

The **heartbeat.pl** utility is used to generate a ShakeCast event XML with event type as “HEARTBEAT.” The output is injected into the ShakeCast system via **sm_inject.pl** and a copy stored in the ShakeMap data directory. This will trigger an event notification to users whom are subscribed to receiving heartbeat events.

The script reads no options from the command line. To create a customized heartbeat event, edit the script located inside the ShakeCast bin directory.

ShakeCast Log File Rotation Tool

NAME

logrotate.pl - ShakeCast Log File Rotation Tool

SYNOPSIS

logrotate.pl [-conf config file]

DESCRIPTION

The **logrotate.pl** utility is used to generate rotating backup files of ShakeCast log files (sc.log, sc_access.log, and sc_error.log). Configurable parameters include rotate-time, max_size, keep-files, compress, and status-file. The administrator can schedule a routine run of this script for maintenance of ShakeCast log files.

The script reads one optional configuration file from the command line. The default configuration file is “sc.conf”.

rotate-time

Specify the time windows for keeping log entries.

max_size

Specify the size limit of log files.

keep-files

Specify the number of backup log files to retain.

compress

Specify the compression option of backup log files.

status-file

Specify the filename of process status.

OPTIONS

–conf

Specify the filename of a custom configuration file to read process parameters for logstats.pl.

ShakeCast Chart Generator for System Statistics

NAME

logstats.pl - ShakeCast Chart Generator for System Statistics

SYNOPSIS

logstats.pl [-conf config file]

DESCRIPTION

The **logstats.pl** utility is used to process ShakeCast log files (sc.log, sc_access.log, and sc_error.log) specified in the system configuration file and generate a set of image files in both histogram and pie charts. The daily activity chart is the default chart displayed in the default page of the Administration Web Interface. The administrator can schedule a routine run of this script to generate new statistics charts.

The script reads one optional configuration file from the command line. The default configuration file is “sc.conf”.

OPTIONS

–conf

Specify the filename of a custom configuration file to read process parameters for logstats.pl.

ShakeCast Event Management Tool

NAME

manage_event.pl - ShakeCast Event Management Tool

SYNOPSIS

manage_event.pl [mode] [option …] event_id [event_id2 …]

DESCRIPTION

The **manage_event.pl** utility is used to re-alert, or delete processed ShakeMap events in the ShakeCast database. It reads one or more event ids from the command line. Mode is one of –resend or –delete. manage_event.pl will return an error message if you do not specify a mode.

–resend

Reprocess notifications for the ShakeMaps and resend notifications to users who are on the recipient list.

–delete

Delete existing events. All information for the processed ShakeMaps will be removed from the ShakeCast database but not downloaded products in the file system.

OPTIONS

–verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

ShakeCast Facility Management Tool

NAME

manage_facility.pl - ShakeCast Facility Management Tool

SYNOPSIS

manage_facility.pl [mode] [option …] file.csv [file2.csv …]

DESCRIPTION

The **manage_facility.pl** utility is used to insert, update, or delete facility data in the ShakeCast database. It reads data from one or more CSV format files. One or more files must be given on the command line. Multiple files can have different formats. Mode is one of –insert, –replace, –delete, –update, or –skip. manage_facility.pl will operate in replace mode if you do not specify a mode.

–insert

New facility records are inserted. It is an error for the facility to already exist; if it does the input record is skipped.

–replace

New records are inserted. If there is an existing facility it is first deleted, along with any associated attributes and fragility levels. All required facility fields must be supplied.

–delete

Delete existing facilities. All required facility fields must be supplied.

–skip

New facility records are inserted. Records for existing facilities are skipped without generating an error. The summary report will indicate how many records were skipped.

–update

Update existing facilities. If the facility does not already exist an error is issued and the record is skipped.

In this mode the only required fields are EXTERNAL_FACILITY_ID and FACILITY_TYPE. Any group values are simply added to the existing set of attributes for the facility, unless the new value matches an existing value, in which case the group value is skipped. For metrics, any metric that appears in the input will be completely replaced.

OPTIONS

–verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

–limit=** n****

Terminate the import after **n** errors in input records. Set to 0 to allow an unlimited number of errors.

This limit only applies to errors encountered when processing a data record from the input file. More serious errors, such as omitting a required field, will always cause the entire input file to be skipped.

–quote=** x****

Use *x* as the quote character in the input file. The default quote character is a quote (“). This character is also used as the escape character within a quoted string.

–separator=** x****

Use *x* as the field separator character in the input file. The default separator character is a comma (,).

FILE FORMAT

manage_facility.pl reads from one or more CSV-formatted files. By default fields are separated by commas and field values that include commas are protected by enclosing them in quotes, but these defaults can be modified; see the **–quote** and **–separator** options below.

The first record in the input file must contain column headers. These headers tell manage_facility.pl how to interpret the rest of the records. Each header field must specify a facility field, a facility metric field, or a group field. The header fields are case-insensitive; facility_name and FACILITY_NAME are equivalent. Fields can appear in any order.

Facility Fields

The following facility names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

external_facility_id (Text(32), required always)

This field identifies the facility. It must be unique for a facility type but the same external_facility_id may be used for different types of facilities.

facility_type (Text(10), required always)

This field identifies the type of facility. It must match one of the types in the facility_type table. Currently defined types are: BRIDGE, CAMPUS, CITY, COUNTY, DAM, DISTRICT, ENGINEERED, INDUSTRIAL, MULTIFAM, ROAD, SINGLEFAM, STRUCTURE, TANK, TUNNEL, UNKNOWN, and HAZUS building types. Refer the HAZUS Damage Level document for the 128 HAZUS building types and code era.

facility_name (Text(128), required for insert/replace)

The value of this field is what the user sees.

short_name (Text(10), optional)

The value of this field is used by ShakeCast when a shorter version of the name is needed due to space limitations in the output.

description (Text(255), optional)

You can use this field to include a short description of the facility.

lat (Float, required for insert/replace)

Specifies the latitude of the facility in degrees and fractional degrees.

lon (Float, required for insert/replace)

Specifies the longitude of the facility in degrees and fractional degrees.

Fragility Fields

Each field beginning with METRIC: is taken to be a facility fragility specifier. The format of a fragility specifier is:

METRIC: *metric-name* **:** *damage-level*

where *metric-name* is a valid Shakemap metric (MMI, PGV, PGA, PSA03, PSA10, or PSA30) and *damage-level* is a valid damage level (GREEN, YELLOW, ORANGE, or RED). Examples of Facility Fragility column labels are METRIC:MMI:RED and metric:pga:yellow.

The metric-name values are defined by the ShakeMap system, and are generally not changed. The above values are current as of summer 2007. The damage-level values shown above are the default values shipped with ShakeCast. These values are defined in your local ShakeCast database, and you may use the administration web interface to change those values and the color-names that refer to them.

Attribute Fields

A facility can have attributes associated with it. These attributes can be used to group and filter facilities.

Each field beginning with ATTR: is taken to be a facility attribute specifier. The format of a facility attribute specifier is:

ATTR: *attribute-name* **:** *attribute-value*

where *attribute-name* is a string not more than 20 characters in length.

Examples of Facility Attribute column labels are ATTR:COUNTY and ATTR:Construction. Attribute values can be any string up to 30 characters long.

EXAMPLES

Example 1 – Point Facilities

Assume we have a file named *ca_cities.csv* containing California cities that we want to load into the ShakeCast database. The file is in CSV format and includes the name of each city and the latitude/longitude of its city center or city hall. Records in the file are of the form

Rancho Cucamonga,34.1233,-117.5794

Pasadena,34.1561,-118.1318

The file is missing two required fields, external_facility_id and facility_type. Since the city name is unique we can add a new column that is a copy of the name column and use that as the external_facility_id. Another column containing the value CITY for each row is added for the facility_type. You can either make these changes using a spreadsheet program or with a simple script written in a text processing language like Perl.

After making these modifications the records look like

CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794

CITY,Pasadena,Pasadena,34.1561,-118.1318

The input file also needs a header record; after adding one the input file looks like

FACILITY_TYPE,EXTERNAL_FACILITY_ID,FACILITY_NAME,LAT,LON

CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794

CITY,Pasadena,Pasadena,34.1561,-118.1318

…

The facilities in this file can now be loaded into ShakeCast using the command

manage_facility.pl ca_cities.csv

Example 2 – Fragility Parameters

It is easy to load fragility parameters for your facilities using **manage_facility.pl**. Building on the previous example, assume a simple model where Instrumental Intensity (MMI) above 7 corresponds to high-level alert (RED), MMI between 5 and 7 corresponds to medium-level alert (YELLOW), and MMI below 5 corresponds to little a low-level (GREEN). The lower threshold of each range (1, 5, 7) is appended to every record in the input file and the header record is changed to reflect the added fields:

FACILITY_TYPE,EXTERNAL_FACILITY_ID,FACILITY_NAME,LAT,LON, \

```
    METRIC:MMI:GREEN,METRIC:MMI:YELLOW,METRIC:MMI:RED
```

CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794,1,5,7

CITY,Pasadena,Pasadena,34.1561,-118.1318,1,5,7

…

Import this file as before. New facility data will replace existing ones.

Example 3 – Multiple Attributes and Multiple Metrics

You can include multiple attributes, multiple metrics, or multiple attributes and multiple metrics for each row of an import file. For example,

FACILITY_TYPE,EXTERNAL_FACILITY_ID,ATTR:COUNTY, ATTR:SIZE,\

```
    METRIC:MMI:GREEN, METRIC:MMI:YELLOW, METRIC:MMI:RED
```

CITY,Rancho Cucamonga,San Bernardino,Small,1,2,6

CITY,Pasadena,os Angeles,Medium,1,2,6

This file would be loaded using the command

manage_facility.pl –update city_county.csv

The above example updates the existing city locations to associate them with a county attribute and a size attribute, and defines the green, yellow, and red alert level shaking thresholds.

ShakeCast Group Management Tool

NAME

manage_group.pl - ShakeCast Group Management Tool

SYNOPSIS

manage_group.pl [mode] [option …] [profile.conf] [lat,lon …]

DESCRIPTION

The **manage_group.pl** utility is used to insert, update, or delete groups in the ShakeCast database and to associate facilities within the profile boundaries with the geometric profile. It reads data from a group configuration file or lat/lon pairs of a polygon from the command line. Mode is one of –insert, –delete, –update, or –poly. manage_group.pl will operate in replace mode if you do not specify a mode.

–insert

New groups are inserted. It is an error if the group already exists; if it does the input record is skipped.

–delete

Delete existing groups. All required group fields must be supplied.

–poly

Read polygon data from the command line and output facility data within the polygon boundaries.

OPTIONS

–conf

Specify the optional profile configuration file.

–verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

FILE FORMAT

manage_group.pl reads data from a file in Apache config format. Lines begin with '#' and empty lines will be ignored. Spaces at the beginning and the end of a line will also be ignored as well as tabulators. If you need spaces at the end or the beginning of a value you can use apostrophe **"**. An option line starts with its name followed by a value. An '=' sign is optional. Some possible examples:

```
    user    max

    user  = max

    user            max
```

If there is more than one statement with the same name, it will create an array instead of a scalar.

Each group is defined as a **block** of options. A **block** looks much like a block in the apache config format. It starts with < **blockname** > and ends with </ **blockname** >. An example:

POLY 35.8000 -116.4000 \ 34.0815 -114.4717 \ 32.0000 -114.3333 \ 32.0000 -120.5000 \ 34.5000 -121.2500 \ 37.2167 -118.0167 \ 36.6847 -117.7930 \ 35.8000 -116.4000 NOTIFICATION_TYPE NEW_EVENT DELIVERY_METHOD EMAIL_HTML EVENT_TYPE ALL NOTIFICATION_TYPE NEW_PROD DELIVERY_METHOD EMAIL_HTML PRODUCT_TYPE GRID_XML EVENT_TYPE ALL

Group Tag Names

The following profile tag names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

poly (float pairs, required always)

This field identifies the boundaries of the profile geometry. It must contain at least three anchor points in order to define a polygon. The total number of anchor points should limit to less than 100, otherwise the administration interface may not be able to display the entire polygon during editing. The manage_profile.pl will however process the polygon definition.

notification (Text(32), optional)

One notification block represents one notification request associated with the profile and applies to all facilities within the profile polygon. Multiple notification blocks for a profile are permitted.

Notification Tag Names

Each notification block defines one notification request. Tag names are corresponding to the field names of the table “profile_notification_request.” Required tags for a notification block include NOTIFICATION_TYPE, DELIVERY_METHOD, and EVENT_TYPE. Valid notification types are CAN_EVENT, NEW_EVENT, UPD_EVENT, SHAKING, NEW_PROD, and DAMAGE.

can_event

This notification request is triggered when an event is cancelled by the seismic network region in which the event was located and the ShakeMap removed from the USGS web site. Require EVENT_TYPE and DELIVERY_METHOD tags.

new_event

This notification request is triggered when an event is located by a seismic network region and a ShakeMap becomes available on the USGS web site. Require EVENT_TYPE and DELIVERY_METHOD tags.

upd_event

This notification request is triggered when the source parameters of an event is updated with a new version by the seismic network. New versions of ShakeMaps for the event may or may not coincide with an updated event. Require EVENT_TYPE and DELIVERY_METHOD tags.

new_prod

This notification request is triggered when a specified ShakeMap product of an event is available on the USGS web site. Require EVENT_TYPE,DELIVERY_METHOD, and PRODUCT tags.

shaking

This notification request is triggered when the ground shaking parameter at the location of the facility exceeds the preset value. Require EVENT_TYPE,DELIVERY_METHOD, METRIC, and LIMIT_VALUE tags.

damage

This notification request is triggered when the ground shaking parameter at the location of the facility falls between the high and low values of facility fragility settings. Require EVENT_TYPE,DELIVERY_METHOD, and DAMAGE_LEVEL tags.

ShakeCast User Management Tool

NAME

manage_user.pl - ShakeCast User Management Tool

SYNOPSIS

manage_user.pl [mode] [option …] file.csv [file2.csv …]

DESCRIPTION

The **manage_user.pl** utility is used to insert, update, or delete user data in the ShakeCast database. It reads data from one or more CSV format files. One or more files must be given on the command line. Multiple files can have different formats. Mode is one of –insert, –replace, –delete, –update, or –skip. manage_user.pl will operate in replace mode if you do not specify a mode.

–insert

New user records are inserted. It is an error for the user to already exist; if it does the input record is skipped.

–replace

New records are inserted. If there is an existing user it is first deleted, along with any associated delivery addresses, notification requests and profiles. All required user fields must be supplied.

–delete

Delete existing users. All required user fields must be supplied.

–skip

New user records are inserted. Records for existing users are skipped without generating an error. The summary report will indicate how many records were skipped.

–update

Update existing users. If the user does not already exist an error is issued and the record is skipped.

In this mode the only required fields are USERNAME and USER_TYPE. Any delivery methods, profiles and users for cloning that appears in the input will be completely replaced.

OPTIONS

–verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

–limit=** n****

Terminate the import after **n** errors in input records. Set to 0 to allow an unlimited number of errors.

This limit only applies to errors encountered when processing a data record from the input file. More serious errors, such as omitting a required field, will always cause the entire input file to be skipped.

–quote=** x****

Use *x* as the quote character in the input file. The default quote character is a quote (“). This character is also used as the escape character within a quoted string.

–separator=** x****

Use *x* as the field separator character in the input file. The default separator character is a comma (,).

FILE FORMAT

manage_ user.pl reads from one or more CSV-formatted files. By default fields are separated by commas and field values that include commas are protected by enclosing them in quotes, but these defaults can be modified; see the **–quote** and **–separator** options below.

The first record in the input file must contain column headers. These headers tell manage_user.pl how to interpret the rest of the records. Each header field must specify a user name field and a user type field. The header fields are case-insensitive; username and USERNAME are equivalent. Fields can appear in any order.

User Fields

The following facility names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

username (Text(32), required always)

This field identifies the user. It must be unique for a user type.

user_type (Text(10), required always)

This field identifies the type of use. It must match one of the types in the user_type table. Currently defined types are: ADMIN, USER, and SYSTEM.

full_name (Text(32), optional)

The value of this field is the user’s full name.

email_address (Text(10), optional)

The value of this field is the user’s email address for receiving communication from the ShakeCast system.

password (Text(10), optional)

The value of this field is used by ShakeCast to generate password for accessing the ShakeCast interface and the web site if password protected.

phone_number (Text(255), optional)

You can use this field to include a user's phone number.

Delivery Method Fields

Each field beginning with DELIVERY: is taken to be a delivery method specifier. The format of a delivery method specifier is:

DELIVERY: *delivery-method*

where *delivery-method* is a valid message format (PAGER, EMAIL_HTML, or EMAIL_TEXT). Examples of Delivery Method column labels are DELIVERY:EMAIL_HTML and delivery:email_html.

The message format values are defined by the ShakeCast system, and are generally not changed. The damage-level values shown above are the default values shipped with ShakeCast. These values are defined in your local ShakeCast database, and you may use the administration web interface to change those values and the color-names that refer to them.

Profile Fields

A user can have notification requests replicated from an existing profile. Each field beginning with PROFILE: is taken to be a profile specifier. The format of a profile specifier is:

PROFILE: *profile-name*

where *profile-name* is a valid profile name.

User Fields

A user can have notification requests replicated from an existing user. Each field beginning with USER: is taken to be a user specifier. The format of a user specifier is:

USER: *shakecast-user*

where *shakecast-user* is a valid user id.

ShakeCast Image Tile Generation Tool

NAME

map_tile.pl – ShakeCast Image Tile Generation Tool

SYNOPSIS

map_tile.pl -type map_type [option …]

DESCRIPTION

The **map_tile.pl** utility is used to generate images tiles to be used by the mapping engine of the web interface. Initially, the generated map tiles are used by the Google Maps API and can also be used by other mapping engines, such as the OpenLayers or OpenStreetMap. It reads one map_type from the command line. The map type includes earthquake, facility, and station.

Earthquake tiles are dynamically updated as soon as a new event is processed by the ShakeCast system. Both facility and station tiles are considered semi-static. Update of these tiles can be done manually or by creating a cron job in the database to schedule generation of tiles.

OPTIONS

–map_type

Specify type of map tile to process. The type must be either “event_tile”, “facility_tile”, or “station_tile”.

–min_zoom

Specify the minimum zoom level to process. The zoom level must be between 1 and 18.

–max_zoom

Specify the maximum zoom level to process. The zoom level must be between 1 and 18.

–rebuild

Delete all existing map tiles of the specified type before generating new map tiles.

–id

Create map tiles only for facility of the selected ID.

–verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

ShakeCast PDF Report Generation Tool

NAME

sc_pdf.pl – ShakeCast PDF Report Generation Tool

SYNOPSIS

sc_pdf.pl -event event_id –version [option …]

DESCRIPTION

The **sc_pdf.pl** utility is used to generate PDF reports for the selected earthquake. It reads one event id and one version from the command line. It will loop through all defined PDF templates in the PDF template directory. If a PDF report is successfully created, it will be registered as a local product and saved into the earthquake-specific data directory.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or queued to be handled by the system workers.

OPTIONS

–event

Specify ID of the event to process.

–version

Specify Version of the event to process.

–verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

ShakeMap Grid/Product Injection Tool

NAME

scfeed_local.pl – ShakeMap Grid/Product Injection Tool

SYNOPSIS

scfeed_local.pl [-event event_id] [option …]

DESCRIPTION

The **scfeed_local.pl** utility is used to process downloaded ShakeMap products located in the ShakeCast data directory. It reads one event id from the command line and creates XML messages before feeding them to ShakeCast. The injection process triggers the ShakeCast process in the same manner as for a real earthquake with respect to facility damage assessment and user notifications.

The name of an unprocessed ShakeMap must match the name of the event ID. ShakeMaps can be downloaded via the USGS ShakeMap link from the ShakeCast Administration Panel or manually from other sources. It will be renamed with the version number appended to the end of the directory name after **scfeed_local.pl** processed the ShakeMap. Outputs of ShakeCast XML files will also be stored in the same directory.

The script will quit gracefully if the ShakeMap has been processed earlier by the ShakeCast system and as a result no notifications will be delivered. To reprocess a ShakeMap that already exists in the ShakeCast system, the administrator will need to either convert the ShakeMap into a test event or delete the event first. In addition to the Administration Interface, an administrator can use the **tester.pl** utility to convert a ShakeMap to a test event and the **manage_event.pl** utility to delete a ShakeMap. The ShakeCast data directory for the deleted ShakeMap also needs to be removed from the file system before starting the reprocess procedure described earlier.

OPTIONS

–event

Specify ID of the event to process.

–scenario

Treat the ShakeMap as a scenario.

–force_run

Force ShakeCast to process the ShakeMap for events that do not meet the process criteria.

–verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

ShakeMap Retrieval/Injection Tool

NAME

shake_fetch.pl – ShakeMap Retrieval/Injection Tool

SYNOPSIS

shake_fetch.pl -network net_id -event event_id [option …]

DESCRIPTION

The **shake_fetch.pl** utility is used to download and process ShakeMap located on the USGS web site. It reads one network id and one event id from the command line. ShakeMap products on the USGS web site are first downloaded before invoking the scfeed_local process to inject the ShakeMap into ShakeCast. The injection process is exactly the same as the scfeed_local script.

OPTIONS

–network

Specify ID of the network to process.

–event

Specify ID of the event to process.

–scenario

Treat the ShakeMap as a scenario.

–force_run

Force ShakeCast to process the ShakeMap for events that do not meet the process criteria.

–verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

–help

Print a synopsis of program usage and invocation options

ShakeCast General Templating Tool

NAME

template.pl – ShakeCast General Templating Tool

SYNOPSIS

template.pl [option …] –event event_id –template template

DESCRIPTION

The **template.pl** utility is used to generate ShakeCast facility summary for the specified event. The script reads at least one event ID and one template file from the command line. The output file is stored in the ShakeCast data directory for the specified event.

–event=s

Specify ID of the event to process.

–template=s

Specify filename of the template used to generate ShakeCast summary. The template files are located under the ShakeCast “template/xml” directory. The system comes with two default templates. “shakecast.tt” is the template for generating “exposure.xml’ and the “kml.tt” for generating Google Earth kml format XML files.

OPTIONS

–version=n

Specify version number of the event to process.

–output=s

Specify filename of the output of ShakeCast summary. The output directory is the ShakeCast data directory for the specified event.

–help

Print a synopsis of program usage and invocation options

FILE FORMAT

template.pl is based on the Perl Template Toolkit. Please see the Template Manual manpage for the complete reference which goes into much greater details about the features and use of the Template Toolkit.

This section covers a brief summary of the template directives. ShakeCast specific identifiers include exposure, item, and type. Facility specific identifiers include name, latitude, longitude, damage_level, MMI, PGA, PGV, PSA03, PSA10, and PSA30.

GET

Evaluate and print a variable or value.

```
[%   GET variable %]

[%       variable %]

[%       hash.key %]

[%         list.n %]

[%     code(args) %]

[% obj.meth(args) %]

[%  "value: $var" %]
```

CALL

As per GET but without printing result (e.g. call code)

```
[%  CALL variable %]
```

SET

Assign a values to variables.

```
[% SET variable = value %]    # 'SET' also optional

[%     variable = other_variable

       variable = 'literal text @ $100'

       variable = "interpolated text: $var"

       list     = [val, val, val, val, ...]

       list     = [val..val]

       hash     = { var => val, var => val, ... }

%]
```

DEFAULT

Like SET above, but variables are only set if currently unset (i.e. have no true value).

```
[% DEFAULT variable = value %]
```

INSERT

Insert a file without any processing performed on the contents.

```
[% INSERT legalese.txt %]
```

INCLUDE

Process another template file or block and include the output. Variables are localised.

```
[% INCLUDE template %]

[% INCLUDE template  var = val, ... %]
```

PROCESS

As INCLUDE above, but without localising variables.

```
[% PROCESS template %]

[% PROCESS template  var = val, ... %]
```

WRAPPER

Process the enclosed block WRAPPER … END block then INCLUDE the named template, passing the block output in the ‘content’ variable.

```
[% WRAPPER template %]

   content...

[% END %]
```

BLOCK

Define a named template block for subsequent INCLUDE, PROCESS, etc.,

```
[% BLOCK template %]

   content

[% END %]
```

FOREACH

Repeat the enclosed FOREACH … END block for each value in the list.

```
[% FOREACH variable = [val, val, val] %]    # either

[% FOREACH variable = list %]                 # or

[% FOREACH list %]                            # or

   content...

   [% variable %]

[% END %]
```

WHILE

Enclosed WHILE … END block is processed while condition is true.

```
[% WHILE condition %]

   content

[% END %]
```

IF / UNLESS / ELSIF / ELSE

Enclosed block is processed if the condition is true / false.

```
[% IF condition %]

   content

[% ELSIF condition %]

     content

[% ELSE %]

     content

[% END %]

[% UNLESS condition %]

   content

[% # ELSIF/ELSE as per IF, above %]

   content

[% END %]
```

SWITCH / CASE

Multi-way switch/case statement.

```
[% SWITCH variable %]

[% CASE val1 %]

   content

[% CASE [val2, val3] %]

   content

[% CASE %]         # or [% CASE DEFAULT %]

   content

[% END %]
```

MACRO

Define a named macro.

```
[% MACRO name <directive> %]

[% MACRO name(arg1, arg2) <directive> %]

...

[% name %]

[% name(val1, val2) %]
```

FILTER

Process enclosed FILTER … END block then pipe through a filter.

```
[% FILTER name %]                       # either

[% FILTER name( params ) %]             # or

[% FILTER alias = name( params ) %]     # or

   content

[% END %]
```

USE

Load a “plugin” module, or any regular Perl module if LOAD_PERL option is set.

```
[% USE name %]                          # either

[% USE name( params ) %]                # or

[% USE var = name( params ) %]          # or

...

[% name.method %]

[% var.method %]
```

PERL / RAWPERL

Evaluate enclosed blocks as Perl code (requires EVAL_PERL option to be set).

```
[% PERL %]

     # perl code goes here

     $stash->set('foo', 10);

     print "set 'foo' to ", $stash->get('foo'), "\n";

     print $context->include('footer', { var => $val });

[% END %]

[% RAWPERL %]

   # raw perl code goes here, no magic but fast.

   $output .= 'some output';

[% END %]
```

TRY / THROW / CATCH / FINAL

Exception handling.

```
[% TRY %]
     content
   [% THROW type info %]
[% CATCH type %]
     catch content
   [% error.type %] [% error.info %]
[% CATCH %] # or [% CATCH DEFAULT %]
     content
[% FINAL %]
   this block is always processed
[% END %]
```

NEXT

Jump straight to the next item in a FOREACH/WHILE loop.

```
[% NEXT %]
```

LAST

Break out of FOREACH/WHILE loop.

```
[% LAST %]
```

RETURN

Stop processing current template and return to including templates.

```
[% RETURN %]
```

STOP

Stop processing all templates and return to caller.

```
[% STOP %]
```

TAGS

Define new tag style or characters (default: [% %]).

```
[% TAGS html %]

[% TAGS <!-- --> %]
```

COMMENTS

Ignored and deleted.

```
[% # this is a comment to the end of line

   foo = 'bar'

%]

[%# placing the '#' immediately inside the directive

    tag comments out the entire directive

%]
```

ShakeCast Task Schedule Tool

NAME

task_inject.pl – ShakeCast Task Schedule Tool

SYNOPSIS

task_inject.pl task [option …]

DESCRIPTION

The **task_inject.pl** utility is used to manually queue a task into the ShakeCast database. The queued task needs to be recognized by the ShakeCast Dispatcher or it will return a FAILED status.

The script is usually invoked from the administration interface but also can be executed directly. Depending on the task type, additional parameters are read from the command line.

OPTIONS

–task

Specify the type of task to process. Default task type includes ‘comp_gmpe’, ‘logrotate’, ‘logstats’, ‘heartbeat’, ‘gs_json’, ‘maintain_event’, ‘facility_fragility_stat’, ‘facility_regulatory_level’, ‘facility_feature_shaking’, ‘screen_shot’, or ‘map_tile’.

comp_gmpe

Compute theoretical ground motions for facilities of the specified earthquake. Additional event ID is read from the command line.

logrotate

Rotate the ShakeCast log files. No additional parameters are required.

logstat

Generate log statistics plots. No additional parameters are required.

heartbeat

Trigger a ShakeCast heartbeat message. No additional parameters are required.

gs_json

Refresh the USGS earthquake JSON feed and process new earthquakes. No additional parameters are required.

maintain_event

Trigger to maintain the ShakeCast database. Old ShakeMaps without exposure will be purged from the system. No additional parameters are required.

facility_fragility_stat

Trigger the process to compute probabilistic facility fragility. Additional one event ID and one Version parameters are read from the command line.

facility_regulatory_level

Trigger the process to compute exceedance of regulatory levels. This function is specifically design for nuclear power plants. Additional one event ID and one Version parameters are read from the command line. **facility_feature_shaking**

Trigger the process to compute facility feature shaking for the selected earthquake. Additional one event ID and one Version parameters are read from the command line.

screen_shot

Take a screen shot for the selected earthquake and save the output image. Additional one event ID and one Version parameters are read from the command line.

map_tile

Generate image tile overlay to be displayed on the web interface. Additional tile type parameter is read from the command line.

–event

Specify ID and Version of the event to process.

ShakeCast Test Event Tool

NAME

tester.pl – ShakeCast Test Event Tool

SYNOPSIS

tester.pl [option …]

DESCRIPTION

The **tester.pl** utility is used to handle ShakeCast test events and includes conversion, listing, and triggering of test events. The script is usually invoked from the administration interface but also can be executed directly. It reads one process type from the command line.

OPTIONS

–type

Specify the type of action to process. Process type is one of ‘event_menu’, ‘new_test’, ‘create_test’, ‘inject_next’, or ‘inject_first’.

event_menu

Output a list of test events available on the system.

new_test

Output a list of actual events on the system that have not been converted into test events.

create_test

Convert the specified event into a test event that can be triggered locally. Require an additional –key option. A new data directory for the event will be created under the “test_data” directory with the name of event ID and “_scte” postfix.

inject_first

Trigger a ShakeCast process for the specified test event as a new event. Require an additional –key option.

inject_next

Trigger a ShakeCast process for the specified test event as an updated event. Require an additional –key option.

–key

Specify ID of the event to process. All information for the processed ShakeMaps will be removed from the ShakeCast database but not downloaded products in the file system.

[]:

Tags: troubleshooting (tag_troubleshooting) shakecast_v3 (tag_shakecast_v3)

©2016 U.S. Geological Survey. All rights reserved.
Page last updated: July 3, 2016
Site last generated: Nov 15, 2016