



**System for Performance Evaluation of Broadband Services
(SPEBS)**

Installation and Administration Manual

Version 1.4

English translation by Dr Nikolaos Tsarmpopoulos
EETT

December 2011

Software Installation

The detailed requirements of SPEBS in terms of hardware and software are described in the respective working group document. A description of the most important installation, configuration and administration procedures, assuming they are performed on a local FreeBSD 8.x server with typical configuration, follows. These procedures should be applicable to any other open source operating systems (e.g. Linux). In parallel, we provide a brief description of how each software module affects other modules.

Required software libraries for the mlab-listen

The statistics collector requires the installation of the following Perl Modules:

1. DBD::Mysql (for communication with the Database)
2. JSON (for deserialization of messages)
3. Crypt::Rijndael (for decryption of the user-id)
4. Log::Log4perl (for logging functionality)
5. Mime::Base64 (for decryption of the user-id)

It is highly recommended that these modules are installed using the typical procedure of FreeBSD ports, so that future upgrade or modification is facilitated. The cpan program may also be used, but may prove harder compared to the first approach.

Database Initialization

In order to record NDT measurements, as well as to support the various SPEBS user interface functions, MySQL version 5.1.54 or newer is required. It is highly recommended that it is installed using the FreeBSD ports procedures. A description of the database schema can be found in the respective document (see file *spebs_schema.sql* in sources).

The database consists of the following tables:

User tables

user connection user_connection rememberme_session access_logs	Used to store user, connection and session info.
--	--

ISP and local exchange tables

isp ipv4_to_isp local_exchange	Used to store ISP and local exchange info. These tables should be manually populated and updated. The inet_aton MySQL function can be used to fill the ip_start and ip_stop columns in ipv4_to_isp table. E.g. if you have the 147.102.0.0/16 range, ip_start would be inet_aton('147.102.0.0') = 2472935424 and ip_stop would be inet_aton('147.102.255.255') = 2473000959.
--------------------------------------	---

The SHAPE column in local_exchange table uses the MySQL geometry type, see: <http://dev.mysql.com/doc/refman/5.5/en/spatial-extensions.html>

Geodata tables

detailed_periphery_polygons	Used to store geographical data, like lists of administrative divisions (peripheries, prefectures, municipalities and postal codes) and their polygons.
detailed_prefecture_polygons	
municipalities	These tables should be manually populated and updated.
peripheries	
peripheries_multipolygons	
postal_codes	
prefecture_polygon	
prefectures	All columns named SHAPE uses the MySQL geometry type, see http://dev.mysql.com/doc/refman/5.5/en/spatial-extensions.html . Longitude and latitude columns refer to the “administrative” center of each (e.g. the capital city of a prefecture).
prefectures_multipolygons	
tk	

Measurement tables

generic_measurement	Used to stored measurement NDT and Glasnost results.
generic_measurements_stats	
glasnost_measurement	
glasnost_measurements_stats	
web100_measurement	
ndt_measurement	

Aggregation tables

aggregation_per_connection	These tables are used to store aggregated measurement data and are populated automatically by the spebs_update.php script via cron.
aggregation_per_connection_glasnost	
aggregation_per_municipality	
aggregation_per_municipality_glasnost	
aggregation_per_periphery	
aggregation_per_periphery_glasnost	
aggregation_per_postal_code	
aggregation_per_postal_code_glasnost	
aggregation_per_prefecture	
aggregation_per_prefecture_glasnost	

To regenerate the database, use the following command line:

```
% mysql -p < spebs_schema.sql
```

Once the database has been created, the mysql user **mlab-listen**) must be given the required access rights, so that the statistics collection daemon can perform its function:

```
GRANT SELECT ON `spebs`.* TO 'mlab-listen'@'localhost';
GRANT INSERT ON `spebs`.`ndt_measurement` TO 'mlab-listen'@'localhost';
GRANT INSERT ON `spebs`.`web100_measurement` TO 'mlab-listen'@'localhost';
GRANT INSERT ON `spebs`.`measurement` TO 'mlab-listen'@'localhost';
GRANT INSERT ON `spebs`.`generic_measurement` TO 'mlab-listen'@'localhost';
```

The corresponding grants for the web application (mysql user **www**) are:

```
GRANT SELECT ON `spebs`.* TO 'www'@'localhost';
GRANT INSERT ON `spebs`.connection TO 'www'@'localhost';
```

```

GRANT UPDATE ON `spebs`.connection TO 'www'@'localhost';
GRANT INSERT ON `spebs`.users TO 'www'@'localhost';
GRANT UPDATE ON `spebs`.users TO 'www'@'localhost';
GRANT INSERT ON `spebs`.user_connection TO 'www'@'localhost';
GRANT UPDATE ON `spebs`.user_connection TO 'www'@'localhost';
GRANT INSERT ON `spebs`.access_logs TO 'www'@'localhost';
GRANT INSERT ON `spebs`.postal_codes TO 'www'@'localhost';
GRANT UPDATE ON `spebs`.postal_codes TO 'www'@'localhost';

```

For the application statistics to work, the application requires the creation and frequent update of the respective database tables. The script `spebs_update.php` should be invoked to run periodically. This could be achieved using `crontab`. A 5 mins interval would be ideal, assuming the system is capable of processing the data between two consecutive invocations:

```
*/5 * * * * php /usr/local/etc/spebs_update.php > /tmp/spebs_update.out 2>&1
```

Installation and Configuration of the mlab-listen daemon

The `mlab-listen` Perl script can be installed in `/usr/local/mlab-listen`. The `mlab-listen.logger` file can be used to configure the logging functionality, i.e. the destination and level of detail:

```
log4perl.logger=DEBUG, LOGFILE
```

There are two levels, `DEBUG` or `WARNING`, to choose from, resulting to different level of daemon logging details.

To configure the location where the log file is stored:

```
log4perl.appender.LOGFILE.filename = /var/log/mlab-listen.log
```

It is highly recommended that the log file is periodically compressed and recycled.

Configure the user authentication details for `mlab-listen` by modifying the variables `db_driver`, `db_base`, `db_host`, `db_user` and `db_password` in the `mlab-listen.conf` file. For example:

```

my $settings = {
    # database type
    db_driver => 'mysql',
    # database name
    db_dbase => 'your_database_name_here',
    # database host
    db_host => 'your_database_host',
    # database host
    db_user => 'your_database_user',
    # database user
    db_passwd => 'your_db_password',
    # how big can the received JSON structure be, in bytes
    read_limit => 5000,
    # after how many seconds do we give up?
    timeout => 30, # after how many seconds do we give up?
    # encryption key for the userid encryption
    key => 'make up an encryption key for the userid encryption',
    # leave this like that, regular expression used to extract the user_id
    pattern => 'random (\d+) string'
};

```

Enable the daemon, via inetd, by adding to `/etc/inetd.conf` the following line:

```
mlab-listen stream tcp nowait/10/3/1 nobody /root/mlab-listen/mlab-listen
```

Also, add to `/etc/services` the following line:

```
mlab-listen 33001/tcp
```

For security reasons, the daemon runs under the nobody user. Configure the file with the appropriate permissions, so that inetd successfully invokes the daemon:

```
% chmod a+x mlab-listen
% chmod 0400 mlab-listen.conf
% chown nobody mlab-listen.conf
```

To test it, restart inetd and telnet:

```
% telnet localhost mlab-listen
```

The daemon should be listening on its port and related messages should be logged.

For security reasons, when modifying the daemon, make sure that the first Perl argument is `-T` so that the script runs under taint mode.

Web App Installation and Configuration

To deploy the web application, the Apache server (version 2.2.17 or newer) and the PHP (version 5.3.5 or newer) Apache HTTPD module (`mod_php`) are required. Installation via the respective FreeBSD port should be sufficient. In addition, the following FreeBSD ports for PHP libraries are required:

<code>php5-bz2</code>	The bz2 shared extension for php
<code>php5-ctype</code>	The ctype shared extension for php
<code>php5-curl</code>	The curl shared extension for php
<code>php5-dom</code>	The dom shared extension for php
<code>php5-filter</code>	The filter shared extension for php
<code>php5-gd</code>	The gd shared extension for php
<code>php5-hash</code>	The hash shared extension for php
<code>php5-iconv</code>	The iconv shared extension for php
<code>php5-json</code>	The json shared extension for php
<code>php5-mbstring</code>	The mbstring shared extension for php
<code>php5-mcrypt</code>	The mcrypt shared extension for php
<code>php5-mhash</code>	The mhash shared extension for php
<code>php5-mysql</code>	The mysql shared extension for php
<code>php5-mysqli</code>	The mysqli shared extension for php
<code>php5-openssl</code>	The openssl shared extension for php
<code>php5-pcre</code>	The pcre shared extension for php
<code>php5-pdo</code>	The pdo shared extension for php
<code>php5-posix</code>	The posix shared extension for php
<code>php5-session</code>	The session shared extension for php
<code>php5-simplexml</code>	The simplexml shared extension for php
<code>php5-soap</code>	The soap shared extension for php

php5-spl	The spl shared extension for php
php5-tokenizer	The tokenizer shared extension for php
php5-xml	The xml shared extension for php
php5-xmlreader	The xmlreader shared extension for php
php5-xmlwriter	The xmlwriter shared extension for php
php5-xsl	The xsl shared extension for php
php5-zip	The zip shared extension for php
php5-zlib	The zlib shared extension for php

SPEBS web app is written in PHP, hence, installation is as simple as saving the application code into a PHP-enabled web content subdirectory. For example, assuming the code is located in `/var/www/spebs`, the following virtual host definition would be sufficient in Apache's configuration file:

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerName    spebs.domain.tld

    Redirect /terms/      http://spebs.domain.tld/?l=0&action=terms
    Redirect /terms      http://spebs.domain.tld/?l=0&action=terms
    Redirect /terms_en/  http://spebs.domain.tld/?l=1&action=terms
    Redirect /terms_en   http://spebs.domain.tld/?l=1&action=terms
    Redirect /about/     http://spebs.domain.tld/?l=0&action=info
    Redirect /about      http://spebs.domain.tld/?l=0&action=info
    Redirect /about_en/  http://spebs.domain.tld/?l=1&action=info
    Redirect /about_en   http://spebs.domain.tld/?l=1&action=info

    ErrorLog      "/var/log/apache/spebs-error.log"
    CustomLog     "/var/log/apache/spebs-access.log" combined

    DocumentRoot "/var/www/spebs"

    <Directory "/var/www/spebs">
        Order          allow,deny
        Allow          from all

        DirectoryIndex index.php index.html
        Options        -Indexes
        AddType        application/x-httpd-php .php

        php_admin_flag display_errors 0
        php_admin_value open_basedir   /var/www/spebs:/var/www/tmp
        php_admin_value error_log      /var/log/apache/php-errors.log
        php_admin_value error_reporting 22527
        php_admin_value safe_mode      0
        php_admin_value safe_mode_gid  0
    </Directory>

    <Directory "/var/www/spebs/notifier">
        Order          allow,deny
        Allow          from all
        Options        -Indexes +FollowSymLinks
        AddType        application/x-java-jnlp-file .JNLP .jnlp
        AddType        application/x-java-archive .jar
        AddType        application/x-java-archive-diff .jardiff
    </Directory>

</VirtualHost>
```

The aforementioned definition should be placed in the *location* or *directory* tag of Apache's configuration file.

The Web app respective configuration settings, in the form of variables, are defined in the file `init.php`, under the directory tree structure of the application. For example, the database connection settings are:

```
$user      = 'root';  
$password = 'dummy';  
$database = 'spebs';
```

Since MySQL does not support an internal function to calculate medians, the installation of an external library (user defined function) is required. The respective implementation is available from web page <http://mysql-udf.sourceforge.net/> and a FreeBSD port is available (`databases/mysql-udf`). Following the installation of the library on the system, it must be loaded into MySQL, using SQL command:

```
CREATE AGGREGATE FUNCTION median RETURNS REAL SONAME 'udf_median.so';
```

Setting up encryption for user-id

The user-id that is communicated by the Web App to the applet and from the latter to the statistics collection daemon is encrypted. Copies of the encryption key are maintained in files

1. For decryption: `mlab-listen` in the statistics collection daemon (subroutine `decode_id`).
2. For encryption: `library.php` in the web application (`produce_encoded_str`).

In case the encryption key is replaced, it should be replaced in the daemon and in the web app. Notice that the applet handles the user-id as a transparent alphanumeric sequence (string), thus it is not affected.

Server Configuration for the NDT tool

To configure the `mlab-listen` daemon, the web app and the applet, a sequence of coordinated actions is required on these three modules.

1. The file `ndt.jar` must be copied to the path where the Web App is stored, so that it can be retrieved by the end-user browser. Note that the file `ndt.jar` must be digitally signed, to gain access to the required end-user computer resources. Otherwise, the applet is inoperable. When the file is self-signed, the end user is asked whether he/she trusts the certificate.
2. The NDT applet parameters that may require re-configuration are:
 - a. **reportHost:** (mandatory) the location of the `mlab-listen` daemon. This is defined in file `parameters.inc.php`

- b. **reportPort:** (optional, default 33001) the port number where the *mlab-listen* daemon will be listening to. This can be defined in file *n.php*, i.e. in the script of the NDT frame.
- c. **testingServer:** (mandatory) the location of the measurements daemon *web100srv*. It is assigned the value of the variable *\$measurement_server* , which is defined in the file *parameters.inc.php*.

Server Configuration for the Glasnost tool

The Glasnost tool comprises two parts, a Java applet and a PHP script. The Java applet, similarly to the NDT client applet, performs the measurement on the client side. The PHP script takes care of collecting the measurement's parameters and, as soon as the measurement is completed, storing and presenting the measurement's results.

The Glasnost Java Applet is fairly simple in terms of appearance and functionality. SPEBS is using the same version that is used on the M-Lab platform.

The Glasnost PHP script has been modified and integrate with the SPEBS web application. It does not require to be separately installed or configured. The basic parameters can be found in file *parameters.inc.php*, along with the Web App parameters.

Application Configuration File

The web app configuration parameters are concentrated in the file *parameters.inc.php*. More specifically, these are:

1. Address (URL) Configuration

\$relative_path: The relative path of the web application address (URL), excluding the domain name. If no relative path is used, this parameter should be empty.

\$home: A full URL of the application, including the domain name and the *\$relative_path* defined above.

2.Database Configuration

\$dbu: the username the web app shall use to connect to the database

\$dbpsw: the password, corresponding to the *\$dbu* user, that the web app shall use to connect to the database

\$database: The database name

3. Session Configuration

\$rememberme_duration: The lifetime of a user browser cookie that maintains a user session open, when the user has selected the «Remember Me» login option

4. Cryptography

Cryptography is used in relation to the NDT applet parameters and the session (cookie) data, when the user selects the «Remember Me» login option.

\$enc_key: Encryption key

\$enc_phrase: Encryption phrase

5. NDT Configuration

\$measurement_server: The NDT measurement server address (IP/FQDN)

\$report_host: The measurement results collection host (currently the same as the web server)

\$MAXUPLOAD: A global maximum upper limit for the uplink performance, measured in Mbps. The headline speed (max theoretical of the service) of the connection is also effective. In case the measured uplink performance exceeds *\$MAXUPLOAD* or the headline speed, the measurement results (resultset) will not be recorded into the database.

\$MAXDOWNLOAD: A global maximum upper limit for the downlink performance, measured in Mbps. The headline speed (max theoretical of the service) of the connection is also effective. In case the measured downlink performance exceeds *\$MAXDOWNLOAD* or the headline speed, the measurement results (resultset) will not be recorded into the database.

6. Glasnost Configuration

\$glasnost_server: Glasnost measurement server (IP/FQDN)

\$glasnost_repeat: Number of repeats per measurement

\$glasnost_duration: Duration of measurement

7. Headline Speeds

\$bandwidths: The name of a database table containing the different headline speeds (service offerings) available in the market. The uplink throughput is stored in 'u' and is measured in Kbps. The downlink throughput is stored in 'd' and is measured in Kbps. A list with all available headline speeds (service offerings) is presented to the user via the 'Register' and 'Configuration' web pages. Then, the uplink/downlink headline speeds are stored in the database along with the user connection parameters.

8. Statistics Generation Parameters

\$min_measurements_per_user: The minimum number of NDT measurements per connection required, before the connection is represented as a pin on the map and the NDT measurement results are taken into consideration for the calculation of the regional statistics.

\$min_glasnostmeasurements_per_user: The minimum required number of Glasnost measurements per connection required, before the connection is represented as a pin on the map and the NDT measurement results are taken into consideration for the calculation of the regional statistics.

\$glasnost_throttles_accepted_percentage: Maximum % of acceptable throttled connection measurements per Glasnost measurement type, per user. When the actual % of throttled connection measurements exceeds this value, the connection is presented as "throttled".

\$glasnost_throttled_connections_accepted_percentage: Maximum % of acceptable throttled connection measurements per Glasnost measurement type, per region. When the actual % of throttled connection measurements exceeds this value, the region is presented as “throttled”.

\$min_connections_per_postal_code: The minimum number of connections fulfilling the *\$min_measurements_per_user* limit that is required before NDT statistics are presented for the Postal Code Region.

\$min_connections_per_municipality: The minimum number of connections fulfilling the *\$min_measurements_per_user* limit that is required before NDT statistics are presented for the Municipality Region.

\$min_connections_per_prefecture: The minimum number of connections fulfilling the *\$min_measurements_per_user* limit that is required before NDT statistics are presented for the County Region.

\$min_connections_per_periphery: The minimum number of connections fulfilling the *\$min_measurements_per_user* limit that is required before NDT statistics are presented for the (administrative) Periphery Region.

\$min_connections_per_country: The minimum number of connections fulfilling the *\$min_measurements_per_user* limit that is required before NDT statistics are presented for the Country.

\$min_connections_per_postal_code_glasnost: The minimum number of connections fulfilling the *\$min_measurements_per_user* limit that is required before Glasnost statistics are presented for the Postal Code Region.

\$min_connections_per_municipality_glasnost: The minimum number of connections fulfilling the *\$min_measurements_per_user* limit that is required before Glasnost statistics are presented for the Municipality Region.

\$min_connections_per_prefecture_glasnost: The minimum number of connections fulfilling the *\$min_measurements_per_user* limit that is required before Glasnost statistics are presented for the County Region.

\$min_connections_per_periphery_glasnost: The minimum number of connections fulfilling the *\$min_measurements_per_user* limit that is required before Glasnost statistics are presented for the (administrative) Periphery Region.

\$sliding_window_in_days: The number days defining a sliding window starting from *\$sliding_window_in_days* ago, ending today, to filter which measurements should be taken into consideration when calculating group statistics.

\$max_distance_from_exchange_meters: A maximum distance, measured in meters, from the Switching Centre (where ADSL/VDSL equipment is operated) to the connection endpoint, beyond which the maximum theoretical connection speed is ‘not available’.

9. Google Maps και recaptcha API Keys

\$googleMapsKey: An API key for use with Google Maps API V2 that corresponds to the domain name of the web server.

\$recaptchaPublicKey: A Public Key for use with the recaptcha library that corresponds to the domain name of the web server.

\$recaptchaPrivateKey: A Private Key for use with the recaptcha library that corresponds to the domain name of the web server.