Giga Creations Tools
“GCTools”

Created by: J. “Giga” Murphy <giga1699@gmail.com>

README File

http://www.gigacreations.net/

Last Updated: 6. July 2011

Table of Contents

About GCTools

Purpose

GCTools was designed to help web developers create dynamic, feature-rich, applications quickly. It was not intended to be a Content Management System (CMS), a web forum, a photo gallery, a login system, or anything in particular. Instead, GCTools aimed to enable developers to create anything they wanted.

This allowed GCTools to be extremely flexible, and streamlined. The developer only has to include those classes that he actually needs, not an entire library of functions that may, or may not, be needed. This is one of the reasons GCTools has been developed so extensively. The developers have put a lot of time into thinking about what it is that a developer may need, and how can we do it with the most simple implementation available.

History

The very first code of GCTools was written in early September, 2010. The initial idea was not to create a framework, but to develop a management system for an object-oriented design class at the University of Louisville. The initial code was fairly small, and only included a few classes.

At the time, PHP had not really been developed with objects in mind. Most code was small scripts that was reminiscent of old C code. Although objects existed in PHP, they had not been utilized to a large extent. This was the aim of J. “Giga” Murphy during his class... to show his class that PHP could be used as a viable object-oriented language for web development.

Following the class, development came to a halt as other classes took priority. The code was commited to a private Git repository, and was largely forgotten about until late April, 2011. At this time development ramped up.

Within a week more than 4 new classes were created, and much of the older code was optimized. GCTools also gained another developer, M. “Beanyhead” Parker. Not only was a new developer added, but the project was released under the GPL v3 license, and pushed to a public GitHub repository for others to utilize. This also enabled people in the community to add to the project, and help to improve it.

Thus, GCTools was officially born and put into open source development.

Future of GCTools

The future of GCTools at this moment is unknown. Open-source developers are still needed to help contribute to the project, and the project is still lacking some major features.

You may check on the status of GCTools at the project's website, http://www.gigacreations.net/

GCTools file version scheme

The version scheme for individual files uses 3 groups (Version: X.X.X). They are explained as follows:

1. First group
    1. This is the overall file version.
    2. An integer greater than zero (0) indicates that everything in the file has been stabilized at one point. Please refer to the following groups for more information about this.
2. Second group
    1. This indicates how many classes in the current file are stable.
    2. An integer of zero (0) means that no class is fully stable yet.
    3. An integer greater than zero (0) means that this many classes in the file are stable.
    4. This does not provide what classes are currently stable. Please refer to the change log for that information.
3. Third group
    1. This is a revision number.
    2. It is not always updated (everyone forgets things sometimes), and thus shouldn't be relied upon.
    3. Basically, the higher the revision number, the greater likelihood that a particular class is almost stable.

Contributing to GCTools

Contributions to the code of GCTools are always welcome. This is why we have released the code under the GPL v3 license. If you haven't already, feel free to fork our project on the GitHub site.

Additionally, if you'd like to become a long-term contributor to GCTools, please send us an e-mail at webmaster@gigacreations.net with some more information about you so we can add you as a contributor to the GitHub repository.

GCTools Files

autoload.inc.php

Class Variables

There are no class variables

Class Functions

The only function is the “__autoload” function which is a “magic” PHP function that tries to include the right file based on what class you are trying to use. It has not been tested in GCTools, but was developed to try to assist new users in loading the proper files before using a class.

cache.inc.php

Class Variables

1.) cacheDir – This defines the directory to store cached files. It must exist, and be writable.

Class Functions

1. Cache($cacheDir)
   (a) $cacheDir defines the directory to store cached files
   (b) Pre/Post Conditions:
      i. Precondition: The $cacheDir should be defined.
      ii. Postcondition: The cache class is initialized.
   (c) This function initializes the Cache class. It ensures that the cache directory exists, and is writable. If the directory does not exist, or is not writable, an exception is thrown.
2. getCacheDir()
   (a) Pre/Post Conditions:
      i. Precondition: $cacheDir should be set
      ii. Postcondition: Returns the $cacheDir, or FALSE otherwise
   (b) This function provides the user with the location of the cache directory.
3. setCacheDir($dir)
   (a) $dir defines the new location of the directory to store cached files.
   (b) Pre/Post Conditions:
      i. Precondition: $dir should be set, and a writable directory
      ii. Postcondition: Return TRUE on success, and FALSE otherwise
   (c) This function allows the user to change the location of the cache directory after the class has already been initialized.
4. createCache($file)
   (a) $file defines the path to the file to be cached
   (b) Pre/Post Conditions:
      i. Precondition: $file should be a valid file
      ii. Postcondition: Create a cache of the file and return TRUE on success, or FALSE on failure
   (c) This function allows the user to create a cache of a file.

Class example

```
<?php
require_once(“cache.inc.php”);

try {
	$cache = new Cache(“/path/to/my/cache/directory”)
}
catch (Exception $e) {
	//There was some kind of error, so handle it
}

if (!$cache->createCache(“/path/to/my/file”)) {
	//There was an error creating the cache
}

//We're done. The cached file is /path/to/my/cache/directory/file.cache.html
?>
```

computer.inc.php

Class Variables

1. id => Defines a unique ID for the computer system.
2. name => Defines the computer's name (generally the DNS name).
3. mac => Defines the computer's MAC address.
4. ip => Defines the IPv4 address of the computer.
5. ip6 => Defines the IPv6 address of the computer (if needed).
6. osType => Defines the type of operating system.
7. osName => Defines the operating system's name.
8. serial => Defines the serial number for the computer.
9. location => Defines a location for the computer.
10. make => Defines the make (manufacturer) of the computer.
11. model => Defines the model of the computer.
12. cpu => Defines CPU information.
13. ram => Defines RAM information.
14. hdd => Defines hard drive information.
15. licensing => Defines licensing information.
16. notes => Defines additional notes about the computer.

Class Functions

1. Computer()
   (a) Pre/Post-conditions:
      i. Precondition: None
      ii. Postcondition: Sets up the Computer class variables for use
   (b) This function is the class constructor. It sets up all the variables for use.
2. getID()
   (a) Pre/Post-conditions:
      i. Precondition: id should be definied
      ii. Postcondition: Return the ID of the computer, or FALSE otherwise
   (b) This function allows the user to get the unique ID of the computer.
3. setID($newID)
   (a) $newID => Defines a new unique ID for the computer.
   (b) Pre/Post-conditions:
      i. Precondition: $newID should be defined

ii. Postcondition: Set the ID

(c) This function allows the user to set the unique ID for the computer.

4. getName()
   (a) Pre/Post-conditions:
      i. Precondition: name should be defined
      ii. Postcondition: Return the name of the computer, or FALSE otherwise.
   (b) This function allows the user to get the name of the computer.
5. setName($newName)
   (a) $newName => Defines the new name for the computer.
   (b) Pre/Post-conditions:
      i. Precondition: $newName should be defined
      ii. Postcondition: Set the new name of the computer.
   (c) This function allows the user to set the name of the computer.
6. getMac()
   (a) Pre/Post-conditions:
      i. Precondition: mac should be defined
      ii. Postcondition: Return the MAC address of the computer, or FALSE otherwise
   (b) This function allows the user to get the MAC address of the computer.
7. setMac($macAddress)
   (a) $macAddress => Defines the MAC address of the computer.
   (b) Pre/Post-conditions:
      i. Precondition: $macAddress should be defined, and be a valid MAC address
      ii. Postcondition: Set the MAC address of the computer.
   (c) This function allows the user to set the MAC address of a computer.
8. getIP()
   (a) Pre/Post-conditions:
      i. Precondition: ip should be defined
      ii. Postcondition: Return the IP address, or FALSE otherwise
   (b) This function allows the user to get the IP address of a computer.
9. setIP($ipAddr)
   (a) $ipAddr => Defines the IPv4 IP address of the computer.
   (b) Pre/Post-conditions:
      i. Precondition: $ipAddr should be defined, and a valid IPv4 address
      ii. Postcondition: Set the IPv4 address
   (c) This function allows the user to set an IPv4 address for the computer.

Class Example

This class is still under development.

database.inc.php

Class Variables

Database class

1.) dbType defines the type of database that is being used.

2.) dbLoc defines the location of the database. This can be an IP address, a hostname, a file location, etc

3.) dbUser defines the username used to connect to the database, if needed.

4.) dbPass defines the password used to connect to the database, if needed.

5.) dbName defines the name of the database to use.

6.) lastError defines the text of the last error that occurred, if any.

MySQL class

1.) myCon defines the MySQL connection

MSSQL class

This class is currently not functional

PGSQL class

This class is currently not functional.

SQLite class

This class is currently not functional.

Class Functions

Database class

1. (Protected) Database($loc, $user, $pass, $name, $type)
   (a) $loc defines the location of the database. This may be an IP address, a hostname or a location on the server.
   (b) $user defines the username used to login to the database.
   (c) $pass defines the password used in conjunction to the username to log into the server.
   (d) $name defines the default database to begin working with.
   (e) $type defines the type of database you will be working with.
   (f) Pre/Post-conditions:
      i. Precondition: The database location, username, password, name and type should be defined.
      ii. Postcondition: The class will set-up the variables that will be used to connect to the database, and conduct queries.

(g) This sets up the class to perform database operations.

2. hasError()
   (a) Pre/Post-conditions:
      i. Precondition: None
      ii. Postcondition: Returns TRUE if an error has occurred, and FALSE otherwise
   (b) This function will enable the user to check if an error has occurred during a database operation.
3. getLastError()
   (a) Pre/Post-conditions:
      i. Precondition: An error should have occurred.
      ii. Postcondition: Returns the error message the the database gave when the last error occurred.
   (b) This function gets the last error message provided by the database.
4. (protected) resetError()
   (a) Pre/Post-conditions:
      i. Precondition: None.
      ii. Postcondition: Any error is cleared from the class.
   (b) This function is used internally by the class to clear any previous errors that occurred.

MySQL class

1. MySQL($loc, $user, $pass, $name, $errorCallback=NULL)
   (a) $loc defines the location of the MySQL server. This can be either an IP address, or a hostname.
   (b) $user defines the username used to log into the MySQL server.
   (c) $pass defines the password used in conjunction with the username to log into the database.
   (d) $name defines the default database to use to perform database functions.
   (e) $errorCallback defines the callback function to send the error an additional way, if one occurs.
   (f) Pre/Post-conditions:
      i. Precondition: The location of the SQL server, the username, the password and the database name is given.
      ii. Postcondition: The MySQL server is connected to
   (g) This function connects to a MySQL server to perform MySQL functions. It will attempt to load the MySQL libraries, if they are not already loaded. It will also initialize it's parent Database class so you can use all the functionality of the abstract Database class.
2. (Private) connect()
   (a) Pre/Post-conditions:
      i. Precondition: The MySQL class should be set up properly.
      ii. Postcondition: The connection to the MySQL server is made, or errors or handled.
   (b) This function is the one that actually makes the connection to the MySQL database.

3. (Protected) throwError([$specialError])
   (a) $specialError defines a unique error that MySQL may not handle by itself, or an error that occurs before a MySQL connection is established
   (b) Pre/Post-conditions:
      i. Precondition: An error should have occurred
      ii. Postcondition: The error is created in the Database class with the proper information.
   (c) This function is called internally when any error has occurred during MySQL operations.
4. query($qString)
   (a) $qString defines the SQL Query string to be executed.
   (b) Pre/Post-conditions:
      i. Precondition: A query should be presented
      ii. Postcondition: The class will attempt to execute the query, and handle any errors.
   (c) SECURITY NOTE:
      i. The user is responsible for handling any sort of SQL injection type attacks. This function DOES NOT handle this by itself.
   (d) This function takes a SQL query, and tries to execute it on a MySQL database. It will also handle any errors that occur during the execution of the query.
5. changeDB($dbName)
   (a) $dbName defines the new database to use for queries.
   (b) Pre/Post-conditions:
      i. Precondition: A database name is given
      ii. Postcondition: The class attempts to change the database to use for MySQL operations. Returns TRUE on success, and FALSE on failure.
   (c) This function changes the database that is used to execute queries.
6. escapeString($string)
   (a) $string defines a string to escape using MySQL
   (b) Pre/Post-conditions:
      i. Precondition: A string should be given
      ii. Postcondition: The string is escaped using the current MySQL connection.
   (c) This function escapes a string to be safe in a MySQL query.
7. connected()
   (a) Pre/Post-conditions:
      i. Precondition: None
      ii. Postcondition: Returns TRUE if the MySQL connection is active, and FALSE otherwise
   (b) This function informs the user if the MySQL connection is still active, or not.
8. reconnect()

(a) Pre/Post-conditions:

i. Precondition: None

ii. Postcondition: Will close any current connection, and re-establish a connection to the MySQL server

(b) This function is used to reconnect to a MySQL server. It is used internally if the MySQL connection is lost, and can be used by the user to reconnect using the given credentials at initialization time.

9. setErrorCallback($callback)

(a) $callback => Defines a callback function that is called when a MySQL error occurs

(b) Pre/Post-conditions:

i. Precondition: $callback should be defined

ii. Postcondition: Set the errorCallback. Returns TRUE on success, and FALSE otherwise.

(c) This function allows the user to set/change the error callback after the MySQL class has been initialized.

MSSQL class

This class is still under development, and has not been tested.

PGSQL class

This class is not currently functional.

SQLite class

This class is not currently functional.

Class example

MySQL class

```
<?php
require_once(“database.inc.php”);
require_once(“error.inc.php”);

$errorHandler = new Error(“errorto@mydomain.com”, “errorfrom@somedomain.com”);
$errorHandler->setErrorSubject(“MySQL Error”);

try {
        $mysql = new MySQL(“hostname”, “username”, “password”, “database”, array($errorHandler, “sendError”));
}
```

```
catch (Exception $e) {
	//Something went wrong
	echo $mysql->getLastError();
}

//Let's do a query
$mysql->query("SELECT * FROM `users` WHERE `username`='" . $mysql->escapeString($username) . "'");

//Check if we're still connected
if (!$mysql->connected())
	$mysql->reconnect();
?>
```

MSSQL class

This class is not currently functional.

PGSQL class

This class is not currently functional.

SQLite class

This class is not currently functional.

error.inc.php

Class Variables

(Abstract) GCError Class

1. errorFrom => Defines the e-mail address to display in the “From:” header when sending an error e-mail
2. errorTo => Defines who error e-mails should be sent to
3. errorSubject => Defines the subject for the e-mail when sending error e-mails

GCErrorHandler Class

1. errorHandlerSet => Defines if the GCErrorHandler has been set for errors.
2. exceptionHandlerSet => Defines if the GCErrorHander has been set for exceptions.

Class Functions

(Abstract) GCError Class

1. GCError($from, $to)
   (a) $from => Defines the from address for sending e-mails
   (b) $to => Defines the to address for sending e-mails
   (c) Pre/Post-conditions:
      i. Precondition: The from, and to, address should be defined
      ii. Postcondition: The Error class is set-up.
   (d) This is the constructor for the Error class. It sets up the class to be able to send error messages via e-mail to an individual, or group.
2. setErrorSubject($errorSubject)
   (a) $errorSubject => Defines the subject line for e-mails sent through this class
   (b) Pre/Post-conditions:
      i. Precondition: None
      ii. Postcondition: The subject line is set
   (c) This function sets the subject line for all e-mails sent via this class.
3. sendError($errorMessage)
   (a) $errorMessage => Defines the body of the e-mail
   (b) Pre/Post-conditions:
      i. Precondition: $errorMessage should be defined
      ii. Postcondition: An e-mail is sent to $to, from $from and containing the body $errorMessage
   (c) This function sends an e-mail to the defined “To:” address(es) with the given error message. This

function may be used as the callback function in the MySQL class (as well as others).

GCErrorHandler Class

1. GCErrorHander($to, $from, $subject=NULL)
   (a) $to => Defines the “To:” address for e-mails
   (b) $from => Defiens the “From:” address for e-mails
   (c) (Optional) $subject => Defines the subject for e-mails
   (d) Pre/Post-conditions:
      i. Precondition: $to, and $from, should be defined.
      ii. Postcondition: Set-up the class
   (e) This is the class constructor. It will properly set-up the class, or report an error if there was a problem.
2. setGCErrorGlobally($errors=NULL)
   (a) (Optional) $errors => Defines the types of errors to report
   (b) Pre/Post-conditions:
      i. Precondition: errorHandlerSet should be false (i.e. you haven't called this function yet)
      ii. Postcondition: Set the PHP error hander. Returns TRUE on success, and FALSE otherwise.
   (c) This sets the GCErrorHander to globally handle all PHP errors.
   (d) REFERENCE: http://php.net/manual/en/function.set-error-handler.php
3. unsetGCErrorGlobally()
   (a) Pre/Post-conditions:
      i. Precondition: Error handler should've been set already
      ii. Postcondition: Return PHP error handler to what it was before. returns TRUE on success, and FALSE otherwise
   (b) This function removes the GCErrorHandler from acting on PHP errors.
4. handleError function
   (a) This function is based off of the PHP guidance for handling errors. You will not need to call it yourself. It will automatically send you an e-mail with the error message, and a backtrace on the error.
5. setGCExceptionGlobally()
   (a) Pre/Post-conditions:
      i. Precondition: exceptionHandlerSet should be false (i.e. you haven't called this function yet)
      ii. Postcondition: Set the PHP exception hander. Returns TRUE on success, and FALSE otherwise.
   (b) This sets the GCErrorHander to globally handle all PHP exceptions.
   (c) REFERENCE: http://www.php.net/manual/en/function.set-exception-handler.php
6. unsetGCExceptionGlobally()

(a) Pre/Post-conditions:

i. Precondition: Exception handler should've been set already

ii. Postcondition: Return PHP exception handler to what it was before. returns TRUE on success, and FALSE otherwise

(b) This function removes the GCExceptionHandler from acting on PHP errors.

7. handleException function

(a) This function is based off of the PHP guidance for handling exceptions. You will not need to call it yourself. It will automatically send you an e-mail with the exception message, and a backtrace on the error.

Class Example

GCErrorHandler Class

```
<?php
require_once(“error.inc.php”);


try {
	$error = new GCErrorHandler(“to@errormessage.com”, “from@errorsfrom.com”);
}
catch (Exception $e) {
	//Handle class init error here
}


$error->setErrorSubject(“THERE WAS AN ERROR!”);


$error->sendError(“There was an error in a script. Please go take a look at it.”);


?>
```

file.inc.php

Class Variables

1. fileLoc => Defines the location of the file on the system.
2. fileName => Defines the name of the file, to include any extension.
3. fileMIMEType => Defines the MIME type of the file.
4. fileBuffer => Defines a buffer that contains the file data.
5. fileSize => Defines the size of the file

Class Functions

1. File($file)
   (a) $file => Defines the location of the file on the system.
   (b) Pre/Post-conditoins:
      i. Precondition: $file should be defined, and an actual file on the system.
      ii. Postcondition: The File class is initialized, and ready for use by the user.
   (c) This function initializes the class, and prepares it for use by the user.
2. (private) addFile($file)
   (a) $file => Defines the location of the file on the system.
   (b) Pre/Post-conditions:
      i. Precondition: $file should be defined, and an actual file on the system.
      ii. Postcondition: Pull all needed data about the file, and fill class variables.
   (c) This function is the core function to this class. It is what actually pulls all the information about the file, and fills the class variables.
3. getFileName()
   (a) Pre/Post-conditions:
      i. Precondition: fileName should be set
      ii. Postcondition: Return the file name, or FALSE otherwise
   (b) This function returns the file name to the user.
4. getMIMEType()
   (a) Pre/Post-conditions:
      i. Precondition: fileMIMEType should be defines
      ii. Postcondition: Return the file's MIME type, or FALSE otherwise
   (b) This function provides the MIME type of the file to the user. It can be useful for limiting file uploads to particular MIME types, or utilizing it when sending an e-mail with an attachment.
5. getSize()

(a) Pre/Post-conditions:
    i. Precondition: fileSize should be defined
    ii. Postcondition: Return the file's size (in bytes), or FALSE otherwise
(b) This function provides the user with the file size, in bytes, of the file that has been loaded into the class.

6. getFile()
(a) Pre/Post-conditions:
    i. Precondition: fileBuffer should be defined
    ii. Postcondition: Return the binary file data, or FALSE otherwise
(b) This function provides the binary data to the user, which is great for attaching a file in an e-mail.

7. hadError()
(a) Pre/Post-conditions:
    i. Precondition: None
    ii. Postcondition: Returns TRUE if an error occurred, or FALSE otherwise
(b) This function enables the user to determine if an error occurred during initialization of the class.

8. moveFile($newFile)
(a) $newFile => Defines the new location and/or file name of the file
(b) Pre/Post-conditions:
    i. Precondition: $newFile should be set
    ii. Postcondition: Move the file, and return TRUE if it was a success, or FALSE otherwise
(c) This function allows a user to move a file on the system.

Class Example

```
<?php

require_once(“file.inc.php”);

try {
        $theFile = new File(“/loc/of/file/filename.txt”);
}
catch (Exception $e) {
        //Handle exceptions here
}

//Print information about the file
```

```
echo “File name: “ . $theFile->getFileName() . “<br>\n”;
echo “File size: “ . $theFile->getFileSize() . “ bytes<br>\n”;
echo “File MIME type: “ . $theFile->getMIMEType() . “<br>\n”;

?>
```

ldap.inc.php

Class Variables

LDAP Class

ActiveDirectory Class

Class Functions

LDAP Class

ActiveDirectory Class

Class Example

LDAP Class

ActiveDirectory Class

mail.inc.php

Class Variables

EMail class

1. $mailTo = > Defines an array of “To:” address for e-mails
2. $mailCC => Defines an array of “CC:” address for e-mails
3. $mailBCC => Defines an array of “BCC:” addresses for e-mails
4. $mailFrom => Defines the “From:” address for e-mails
5. $mailReplyTo => Defines the “ReplyTo:” address fro e-mails
6. $mailSubject => Defines the subject of the e-mail
7. $mailMessage => Defines the body of the message
8. $mailAttachments => Defines an array of Attachments for the e-mail
9. $mailAddlHeaders => Defines any additional headers to be sent with the e-mail
10. (private)$mailSplit => Defines a splitting string that is used when sending e-mails with attachments.

Class Functions

EMail class

1. Email()
   (a) Pre/Post-conditions:
      i. Precondition: None
      ii. Postcondition: Set-up the Email class for use by the user
   (b) This function prepares the EMail class for use by the user. It initializes all the variables needed.
2. addTo($address)
   (a) $address => Defines a valid e-mail address
   (b) Pre/Post-conditions:
      i. Precondition: A valid e-mail address is supplied
      ii. Postcondition: Return TRUE if the address was added, and FALSE otherwise
   (c) This function add an e-mail address to the list of “To:” addresses.
3. (private) formatTo()
   (a) This function is used to properly format the “To:” addresses for sending an e-mail using PHP's mail() function.
4. addCC($address)
   (a) $address => Defines a valid e-mail address
   (b) Pre/Post-conditions:

i. Precondition: A valid e-mail address is supplied

ii. Postcondition: Return TRUE if the address was added, and FALSE otherwise

(c) This function add an e-mail address to the list of “CC:” addresses.

5. (private) formatCC()

(a) This function is used to properly format the “CC:” addresses for sending an e-mail using PHP's mail() function.

6. addBCC($address)

(a) $address => Defines a valid e-mail address

(b) Pre/Post-conditions:

i. Precondition: A valid e-mail address is supplied

ii. Postcondition: Return TRUE if the address was added, and FALSE otherwise

(c) This function add an e-mail address to the list of “BCC:” addresses.

7. (private) formatBCC()

(a) This function is used to properly format the “BCC:” addresses for sending an e-mail using PHP's mail() function.

8. setFrom($address)

(a) $address => Defines a valid e-mail address

(b) Pre/Post-conditions:

i. Precondition: A valid e-mail address is provided

ii. Postcondition: The “From:” address is set

(c) This functions enables the user to set the “From:” address for e-mails sent through this class.

9. getFrom()

(a) Pre/Post-conditions:

i. Precondition: The “From:” address should be set

ii. Postcondition: Return the “From:” address, or FALSE otherwise

(b) This function allows the user to see what the currently set “From:” address is set to. If one is not set, the function returns FALSE.

10. setReplyTo($address)

(a) $address => Defines a valid e-mail address

(b) Pre/Post-conditions:

i. Precondition: A valid e-mail address is provided

ii. Postcondition: The “Reply-to:” header is set

(c) This function enables the user to set the “Reply-to:” header for e-mails set out using this class. This allows the user to define a specific “From:” address, but have the default reply action go to the “Reply-to:” address.

11. getReplyTo()

(a) Pre/Post-conditions:

i. Precondition: The “Reply-to:” address should be set

ii. Postcondition: Returns the “Reply-to:” address, or FALSE otherwise

(b) This function allows the user to know the currently set “Reply-to:” address. If one is not currently set, the function will return FALSE.

12. setSubject($subject)

(a) $subject => Defines the subject of the current e-mail

(b) Pre/Post-conditions:

i. Precondition: A subject should be defined

ii. Postcondition: Sets the subject of the current e-mail

(c) This function allows the user to set a specific subject for the e-mail message

13. getSubject()

(a) Pre/Post-conditions:

i. Precondition: A subject should be defied already

ii. Postcondition: Return the subject of the e-mail message, or FALSE otherwise.

(b) This function allows the user to see what the currently set subject is.

14. setMessage($message)

(a) $message => Defines the actual body of the e-mail message

(b) Pre/Post-conditions:

i. Precondition: A message should be definied

ii. Postcondition: Set the message for the e-mail

(c) This function allows the user to set the body of the e-mail message. It may, or may not, contain HTML and/or plain-text.

15. getMessage()

(a) Pre/Post-conditions:

i. Precondition: A message body should be set

ii. Postcondition: Return the message body, or FALSE otherwise

(b) This function allows the user to see what the currently set message body is.

16. addAttachment($file)

(a) $file => Defines the location, and name, of a file on the system to attach to the e-mail message

(b) Pre/Post-conditions:

i. Precondition: A valid file is given

ii. Postcondition: Attaches the given file to the e-mail message

(c) This function allows the user to attach a file from the system to the e-mail to be sent out.

17. setAddlHeaders($headers)

(a) $headers => Defines additional headers to be set

(b) Pre/Post-conditions:

i. Precondition: Valid headers are given

ii. Postcondition: Adds the additional headers to the e-mail

(c) This function allows the user to add additional headers to the e-mail message. This function should not be used to set the “From:” or “Reply-To:” headers.

18. getAddlHeaders()

(a) Pre/Post-conditions:

i. Precondition: Additional headers should be defined.

ii. Postcondition: Return the additional headers, or FALSE otherwise.

(b) This function allows the user to see what additional headers have been set for the e-mail.

19. (protected) getHeaders()

(a) Pre/Post-conditions:

i. Precondition: None

ii. Postcondition: Properly format all additional headers

(b) This function is an internal function to the EMail class. It is used to properly format the headers to be used with the PHP mail() function.

20. send()

(a) Pre/Post-conditions:

i. Precondition: The “To:” address should be set

ii. Postcondition: The e-mail message should be sent. Returns TRUE on success, and FALSE otherwise.

21. validEmail($address)

(a) $address => Defines an e-mail address to check.

(b) Pre/Post-conditions:

i. Precondition: An address is given

ii. Postcondition: Returns TRUE if the given address is valid, and FALSE otherwise.

(c) This function checks if the given e-mail address is a valid address or not. It also does some additional checking to ensure that it can be sent through the various e-mail systems.

Class Example

```
<?php

$email = new EMail();

$email->setFrom(“billing@mycompany.com”);
```

```
$email->addTo(“somebody@somedomain.com”);
$email->addBCC(“myboss@mycompany.com”);
$email->setReplyTo(“me@mycompany.com”);
$email->setSubject(“Your new bill”);

$message = “Dear Sombody,

Attached is your new bill for this month. Please pay it within 15 days.

If you have any questions, please reply to this e-mail.

Thank you,
My Company Billing Services”;

$email->setMessage($message);

$email->addAttachment(“/home/billing/newbills/somebody.new.bill.pdf”);

$email->send();
?>
```

navigation.inc.php

Class Variables

Page class

1. pageID => Defines a unique page ID
2. pageName => Defines the name of the page
3. pageTitle => Defines the title of the page
4. pageURL => Defines the full URL of the page
5. pageContent => Defines the HTML content of the page
6. pageChildren => An array of Page classes that are children of the current page
7. pageLastMod => Defines when the page was last modified (YYYY-MM-DD HH:MM:SS format)
8. pageChangeFreq => Defines how often the page changes
9. pagePriority => Defines the priority level of the page
10. pageEnabled => Defines if the page is enabled, or not
11. pagePHP => Defines if this is a PHP page
12. pageSecurityCheck => Defines a callback to check the security permissions of a user on the given page

Navigation class

1. navPages => An array of Page classes that create the navigation structure

Class Functions

Page class

1. Page()
   (a) Pre/Post-conditions:
      i. Precondition: None
      ii. Postcondition: Set-up the Page class to be ready for use
   (b) This is the constructor for the Page class. It will set-up all needed variables for use by the user.
2. getPageID()
   (a) Pre/Post-conditions:
      i. Precondition: pageID should be defined
      ii. Postcondition: Return the page ID, or FALSE otherwise
   (b) This function allows the user to get the unique ID of the page
3. setPageID($id)
   (a) $id => Defines the new page ID

(b) Pre/Post-conditions:
    i. Precondition: $id should be defined
    ii. Postcondition: Set the page ID. Return TRUE on success, and FALSE otherwise
(c) This function allows the user to set the unique page ID of the page

4. getPageName()
(a) Pre/Post-conditions:
    i. Precondition: pageName should be defined
    ii. Postcondition: Return the name of the page, or FALSE otherwise
(b) This function allows the user to see the name of the page.

5. setPageName($name)
(a) $name => Defines the new name of the page
(b) Pre/Post-conditions:
    i. Precondition: $name should be defined
    ii. Postcondition: Sets the name of the page. Return TRUE on success, and FALSE otherwise.
(c) This function allows the user to set the name of the page.

6. getPageTitle()
(a) Pre/Post-conditions:
    i. Precondition: pageTitle should be defined
    ii. Postcondition: Returns the page title, or FALSE otherwise
(b) This function allows the user to get the title of the page.

7. setPageTitle($title)
(a) $title => Defines the new title for the page
(b) Pre/Post-conditions:
    i. Precondition: $title should be defined
    ii. Postcondition: Set the new page name. Return TRUE on success and FALSE otherwise.
(c) This function allows the user to set the title of the page.

8. getPageURL()
(a) Pre/Post-conditions:
    i. Precondition: pageURL should be set
    ii. Postcondition: Return the page URL, or FALSE otherwise
(b) This function allows the user to set the URL of the page

9. setPageURL($url)
(a) $url => Defines the new page URL
(b) Pre/Post-conditions:
    i. Precondition: $url should be a valid URL

ii. Postcondition: Set the new page URL

(c) This function allows the user to set the page URL

10. getPageContent()

(a) Pre/Post-conditions:

i. Precondition: pageContent should be defined

ii. Postcondition: Returns the content of the page, or FALSE otherwise

(b) This function allows the user to get the content of a page. It will automatically get dynamic content from PHP pages as well so that you have the most up-to-date version of the page.

11. setPageContent($content)

(a) $content

i. Can define the actual content of the page, or...

ii. Can define a PHP page to use as the content

(b) Pre/Post-conditions:

i. Precondition: Valid content is provided, and the user has access to if (if security is being used)

ii. Postcondition: The content is set. Returns TRUE on success and FALSE otherwise

(c) This function allows the user to set the page content, to include dynamic content from PHP pages. The function will also check if the user has rights to the page, if you are using the security callback. If they do not have access to the content, the function will return FALSE just like any other error.

(d) The callback to check for page security utilizes a unique page ID, so if you haven't defined those yet you will need to before using that functionality.

12. hasChildren()

(a) Pre/Post-conditions:

i. Precondition: None

ii. Postcondition: Returns TRUE if the page has children, and FALSE otherwise

(b) This function allows the user to check if the page has any children.

13. getChildren()

(a) Pre/Post-conditions:

i. Precondition: Page should have children

ii. Postcondition: Return children pages, or FALSE otherwise

(b) This function allows the user to get the children of the page.

14. addChild($child)

(a) $child => Defines another Page class that is a child of the present Page class

(b) Pre/Post-conditions:

i. Precondition: A valid Page class is passed to the function

ii. Postcondition: The pageChildren array is updated. Returns TRUE on success, and FALSE otherwise.

(c) This function allows the user to add a child to the current Page class.

15. getLastMod()
    (a) Pre/Post-conditions:
        i. Precondition: pageLastMod should be definied
        ii. Postcondition: Returns the last modified time, or FALSE otherwise.
    (b) This function allows the user to get the last modified time of the page.
16. setLastMod($lastMod)
    (a) $lastMod => Defines a date/time that the page was last modified
    (b) Pre/Post-conditions:
        i. Precondition: $lastMod should be a valid date/time
        ii. Postcondition: Updates the pageLastMod value. Returns TRUE on success, and FALSE otherwise.
    (c) This function allows the user to set the last modified date/time for the page.
17. getChangeFreq()
    (a) Pre/Post-conditions:
        i. Precondition: pageChangeFreq should be defined
        ii. Postcondition: Return the page's change frequency, or FALSE otherwise
    (b) This function allows the user to know the change frequency of the page.
18. setChangeFreq($frequency)
    (a) $freq => Defines a change frequency.
        i. Must be one of the following values:
            A. always
            B. hourly
            C. daily
            D. weekly
            E. monthly
            F. yearly
            G. never
    (b) Pre/Post-conditions:
        i. Precondition: $frequency should be defined, and one of the acceptable values
        ii. Postcondition: Sets the page change frequency. Returns TRUE on success, and FALSE otherwise.
19. getPriority()
    (a) Pre/Post-conditions:
        i. Precondition: pagePriority should be defined

ii. Postcondition: Returns the page's priority, or FALSE otherwise

(b) This function allows the user to know the priority of the page.

20. setPriority($priority)
    (a) $priority => Defines a double between 0 and 1 that show the page's priority
    (b) Pre/Post-conditions:
        i. Precondition: $priority should be defined, and be between 0.0 and 1.0
        ii. Postcondition: Sets the pagePriority value. Returns TRUE on success, and FALSE otheriwse.
    (c) This function allows the user to set the priority of the page.
21. enablePage()
    (a) Pre/Post-conditions:
        i. Precondition: None
        ii. Postcondition: Set pageEnabled to TRUE. Returns TRUE on success, and FALSE on failure.
    (b) This function allows the user to enable the page
22. disablePage()
    (a) Pre/Post-conditions:
        i. Precondition: None
        ii. Postcondition: Sets pageEnabled to FALSE. Returns TRUE on success, and FALSE otherwise.
    (b) This function allows the user to disable the page.
23. isEnabled()
    (a) Pre/Post-conditions:
        i. Precondition: None
        ii. Postcondition: Returns TRUE if the page is enabled, and FALSE otherwise.
    (b) This function allows the user to check if the page is enabled, or not.
24. isPHP()
    (a) Pre/Post-conditions:
        i. Precondition: None
        ii. Postcondition: Returns TRUE if the page is a PHP page, and FALSE otherwise.
    (b) This function allows the user to check if the page is a PHP page, or not.
25. setPHP()
    (a) Pre/Post-conditions:
        i. Precondition: None
        ii. Postcondition: Sets the page to be a PHP one. Returns TRUE on success, and FALSE otherwise.
    (b) This function allows the user to make the page a PHP one.
26. unsetPHP()
    (a) Pre/Post-conditions:

i. Precondition: None

ii. Postcondition: Sets the page to NOT be a PHP page. Returns TRUE on success, and FALSE otherwise.

(b) This function allows the user to remove the status of being a PHP page.

27. setSecurityCallback($callback)
    (a) $callback => Defines a callable function that accepts one input, the unique page ID
    (b) Pre/Post-conditions:
        i. Precondition: $callback should be defined, and callable.
        ii. Postcondition: Set the pageSecurityCheck variable
    (c) This function allows the user to define a callback function to check the security permissions of a given page
28. unsetSecurity()
    (a) Pre/Post-conditions:
        i. Precondition: None
        ii. Postcondition: Remove the security callback. Return TRUE on success, and FALSE otherwise
    (b) This function allows the user to get rid of the security checking.
29. hasSecurity()
    (a) Pre/Post-conditions:
        i. Precondition: None
        ii. Postcondition: Returns TRUE if security is enabled, and FALSE otherwise
    (b) This function allows the user to check if the page has security enabled, or not.

Navigation class

1. Navigation
   (a) Pre/Post-conditions:
       i. Precondition: None
       ii. Postcondition: Sets-up the Navigation class
   (b) This function sets up the Navigation class for the user to use
2. addPage($page)
   (a) $page => Defines a Page class to be added to the navigation structure
   (b) Pre/Post-conditions:
       i. Precondition: $page should be a Page class
       ii. Postcondition: Add the page to the navigation structure
   (c) This function allows the user to add a page to the navigation structure.
3. getPage($pageName)
   (a) $pageName => Defines the name of a page in the navigation structure

(b) Pre/Post-conditions:

i. Precondition: $pageName should be a page that exists

ii. Postcondition: Return the Page class of the page, or FALSE otherwise

(c) This function allows the user to get all the page information for a page based on its name.

4. getHomePage()

(a) Pre/Post-conditions:

i. Precondition: The home page should be set (i.e. a page name of “”)

ii. Postcondition: Return the Page class of the home page

(b) This function allows the user access to the home page of the navigation structure. This is a unique function created specifically for the home page.

5. getSitemap()

(a) Pre/Post-conditions:

i. Precondition: Pages should exist in the navigation structure

ii. Postcondition: Return a XML sitemap based on sitemaps.org standard, or FALSE otherwise.

(b) This function generates a XML sitemap that the user can submit to search engines describing the layout of their website.

6. (protected) addPageToSitemap($page)

(a) This function is internal to the Navigation class, and is used to actually add the pages to the XML sitemap in the getSitemap() function.

7. pageExists($pageName)

(a) $pageName => Defines the name of a page that the user is looking for.

(b) Pre/Post-conditions:

i. Precondition: $pageName should be defined

ii. Postcondition: Returns TRUE if the page exists, and FALSE otherwise

(c) This function allows the user to check if a particular page exists in the navigation structure.

8. formatPageName($name)

(a) $name => Defines a page name

(b) Pre/Post-conditions:

i. Precondition: $name should be given

ii. Postcondition: Format the page name into a “SEO”-friendly format

(c) This function allows the user to format a page name for SEO optimization.

Class Example

photo.inc.php

Class Variables

1. picLoc => Defines the location of the picture on the system
2. picBuffer => Defines a buffer that holds the raw picture data
3. picSize =>

Class Functions

Class Example

security.inc.php

Class Variables

Class Functions

Class Example

session.inc.php

Class Variables

Class Functions

Class Example

user.inc.php

Class Variables

Class Functions

Class Example

Credits

Code Contributors

1.) J. “Giga” Murphy
    1. E-Mail: webmaster@gigacreations.net
    2. Website: http://www.gigacreations.net/
2.) M. “Beanyhead” Parker
    1. E-Mail: beanyhead@gmail.com
    2. Website: http://www.nomits.com/
3.) N. Miles
4.) R. Rios

Change Log

This is in the works

License

GCTools has been released under the GPL v3 license, which is as follows:

GPL v3 License

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you

these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could

make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other

parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all

the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do

not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not

used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated
place (gratis or for a charge), and offer equivalent access to the
Corresponding Source in the same way through the same place at no
further charge. You need not require recipients to copy the
Corresponding Source along with the object code. If the place to
copy the object code is a network server, the Corresponding Source
may be on a different server (operated by you or a third party)
that supports equivalent copying facilities, provided you maintain
clear directions next to the object code saying where to find the
Corresponding Source. Regardless of what server hosts the
Corresponding Source, you remain obligated to ensure that it is
available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided
you inform other peers where the object code and Corresponding
Source of the work are being offered to the general public at no
charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some

trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and

propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of

this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have

permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS