

# **freEcards manual**

## **For version 3.0**

*by MTVWebdesign  
August 2000*

# Table of Contents

<b>Table of Contents</b> .....	<b>2</b>
<b>Preface</b> .....	<b>3</b>
Contact information:.....	3
<b>Getting the script</b> .....	<b>4</b>
<b>What you get with the package</b> .....	<b>4</b>
Script root/ .....	4
<b>Requirements</b> .....	<b>5</b>
<b>Installing the script</b> .....	<b>6</b>
Changes you have to make .....	6
Uploading the files.....	6
Running the setup program.....	6
Uploading the rest of the files.....	7
<b>Installing the plugins</b> .....	<b>8</b>
<b>Installing the Page generator</b> .....	<b>8</b>
<b>Advanced setup</b> .....	<b>8</b>
<b>List of available variables in freecards.cgi</b> .....	<b>9</b>
Normal Setup.....	9
Advanced setup .....	11
Plugins Setup.....	12
FileStore plugin .....	13
Layout plugin .....	13
SendPics plugin .....	13
ToSend plugin .....	14
<b>Creating a postcard desk</b> .....	<b>15</b>
<b>List of variables available in a postcard desk</b> .....	<b>16</b>
Hidden fields .....	16
User defined fields .....	17
<b>Plugin information</b> .....	<b>20</b>
Installation.....	20
FileStore plugin.....	20
Layout Plugin .....	20
SendPics Plugin.....	22
ToSend plugin.....	22
<b>Webmaster options</b> .....	<b>23</b>
<b>Common Problems and solutions</b> .....	<b>24</b>
Languagepacks .....	24
<b>Matching site layout with page generator</b> .....	<b>25</b>

## Preface

Thank you purchasing MTVWebdesign's freEcards script, the perfect script to give you the ability to host an ecard service or simply use the script for personal use.

Version 3 is the result of 4 years of programming. Since version 1 a lot of things changed, making the script perform better, and making it easier to use. Since version 2.4 the script can also handle some plugins.

Version 3 has some minor bug fixes, but the way the script is installed is actually really the only thing that really changed. We had some complaints about the way people needed to install the package. Now you get an admin script, which you simply need to run. It'll ask some questions, but on most questions it'll already know the answer! Simply pressing the Install button on the page will install the script.

The installation script also has support for the page generator script, and the generated page is now easier to change for people using a WYSIWYG HTML editor.

If you are experiencing difficulties and you cannot find the answer in this manual, feel free to email us at: [freecardsbuyer@mtvwebdesign.hypermart.net](mailto:freecardsbuyer@mtvwebdesign.hypermart.net)  
We'll be in contact with you as soon as possible.

## Contact information:

Maarten Veerman  
Balthasar vd Polweg 106  
2628 AW DELFT  
The Netherlands.

## Getting the script

The script is available at:

<http://mtvwebdesign.hypemart.net/scripts/freecards>

Please note that we are working on an auto-update feature for your convenience.

## What you get with the package

Below is the database structure, as it exists in the zip file you downloaded and the way the admin first time installation will install it.

### Script root/

<b>admin.cgi</b>	
<b>dbUnder.lib</b>	
<i>freecards.cgi</i>	
<b>freecards_eng.lib</b>	
<i>getpic.cgi</i>	
<b>mtvwebdesign.lib</b>	
<i>pgen.cgi</i>	
<b>pgen_eng.lib</b>	
<b>databases/</b>	<i>database files</i>
<b>images/</b>	htmlbuttons.js
	blackpix.gif
	logo.gif
	noimg.gif
	pixel.gif
	stamp.gif
<b>languagepacks/</b>	de.lng
	en.lng
	etc.
<b>plugins/</b>	filestore.pl
	layout.pl
	sendpics.pl
	tosend.pl
	<b>layouts/</b>
	<u>NAME_preview.html</u>
	<u>NAME_receive.html</u>
	<i>uploaded image files</i>
<b>terms/</b>	<b>stored_images/</b>
	privacy.txt
	terms.txt

Bold lines are directories.

Italic lines are files/directories, which are automatically created by the script

Underlined lines don't exist yet, but you can create them yourself and upload them.

Please note that all plugins are named here. It might be possible that you do not own all the plugins, but if you do you can upload them to the correct directory.

## Requirements

To run the script you need a server that allows you to run Perl/CGI scripts (and send email). There are a lot of these servers available on the WWW today.

If you are planning to use the `getpic_script`, or if you want to use the SendPics plugin, you need the following modules:

**LWP::Simple**  
**MIME::Base64**

These modules are probably already installed on your server. If not, please contact your webmaster.

# Installing the script

## *Changes you have to make*

Before you can upload the files, you need to make some changes. Unzip the files using WinZip or an other ZIP program, and make sure that you set the option to use the folder names available in the zip file.

After you unzipped all the files, you need to open admin.cgi in an ASCII based editor like NotePad or SimplePad, which are both ideal for the job. You need to change the first line in the admin.cgi file to match the location of Perl on your server. You can get this information from your webmaster.

Example:

If the location of Perl on your server is: /usr/local/bin/perl, the first line in the admin.cgi file will need to look like:

```
#!/usr/local/bin/perl
```

Some servers require you to rename admin.cgi to admin.pl.

## *Uploading the files*

In the first place, you only need to upload the file you find in the root directory of the script, to a directory on your server where you want the script to be. You need to upload the following files:

```
admin.cgi  
dbUnder.lib  
freecards_eng.lib  
htmlbuttons.js  
mtvwebdesign.lib  
pgen_eng.lib  
en.lng
```

Please make sure to upload all these files in ASCII format. See the instructions of your FTP client for more information.

After you uploaded the files, you need to set the permissions on admin.cgi to 755. In some FTP-clients this means you need to change the attributes. Make sure that the owner has read-write-execute permissions, the group only read and execute permissions and the www-user also only read and execute permissions.

You don't have to create any other directories. The script will handle this.

## *Running the setup program*

To run the setup program (admin.cgi), simply type the URL to the script in the location bar of your browser.

The script will run. If you get a 500 Internal Server error, you need to re-upload the file and make sure that this time you do upload in ASCII format, or check the permissions on the file again.

If everything is correct, you'll see a page that tells you that a first time setup is detected. Simply follow the steps and if you have a question, or don't know what we mean with the question, press the "?" at the beginning of the question.

After you checked all the questions, click the install button at the bottom of the page. The setup program will now install the script with some default settings you can change later in the advanced setup page.

### ***Uploading the rest of the files***

After the installation you'll see a page with the database structure the script now needs. The directories are also created. You can now upload the rest of the files to correct location. You can see this directory structure at any time again by clicking the "Directory Structure" link in the admin mainpage.

The script is now installed! By default, no plugins are installed. Please see "Installing the plugins" for more information.

## Installing the plugins

To install the plugins you first need to upload the plugins you own to the correct plugins directory. Make sure to upload in ASCII format.

Then enter the URL to the admin.cgi file in the location bar of your browser. Login and click the link: "Install Plugins".

The script will automatically detect which plugins you have on the server, and will print a nice form for each plugin. Simply answer those questions, many of which are already answered. Then click the submit button and you can now use the plugins!

## Installing the Page generator

Installing the page generator has never been easier than this: simply go to the admin-mainpage and click the link: "Install page generator". You are asked to verify the two questions, and if the script determined them correctly, you simply click the submit button and the page generator script is ready for use!

## Advanced setup

The advanced setup is available for people who want to personalize the setup routines even more. Please note that changing variables in the advance setup page may result in not functioning of the script. Please see the "[List of available variables in freecards.cgi](#)" for further details.



# List of available variables in freecards.cgi

## Normal Setup

### **\$webmasterusername**

This variable is the username you need when accessing the admin script. You also need it when you want to access the database directory.

### **\$webmasterpass**

This is the password you need to access the admin script and the database directory. This password is not encrypted and that isn't really needed too, because there is no way people using the WWW to read your script to see the inside of the script, since it'll always execute.

### **\$webmasteremail**

This is your email address which is used in the welcome email and when mailing active users.

### **\$servicename**

This is the name of the service if you are going to use the script for running a service.

### **\$servicelocation**

This is the URL to a page where people can find information about the service you are running, and how they can get their own postcard desk.

### **\$perl**

This is the location of Perl on your server. Normally this should be something like: `#!/usr/bin/perl` or `#!/usr/local/bin/perl`

### **\$mailprog**

This variable should point to the location of SendMail on your server. Please note that when using the setup script (admin.cgi) you don't need to add the pipe and the `-t` flag, but if you install manually, you do. For example:

```
$mailprog = "|/usr/sbin/sendmail -t";
```

### **\$cgiext**

This is the extension used for the generated files. It can either be `.cgi` or `.pl`

### **\$script\_location**

This should be the URL to the freecards.cgi (or freecards.pl) file on your server. It's very important that this option is correct.

### **\$example\_carddesk**

If you will use the script for a service, you'll need to set this option. It should be the URL to a postcard desk people can visit to have a look at it, and see the possibilities that your service has.

### **\$std\_ing**

This is the standard language. You can use the country code for the language packs. (English is "en", Dutch is "nl", etc.). Please note that you need a matching XX.lng file in the language packs directory. See en.lng for instructions on how to translate a language pack.

### **\$max\_recipients**

A user of your script can send a card to multiple recipients. Due to spam filters you might not want this option, than you should set this option to "1". If you do want the option, "10" is a nice number, because most spam filters won't allow you to send more than 10.

### **\$elapsed\_days**

This is the number of days a card will stay in the database before it's deleted.

### **\$elapsed\_days\_two**

If a card is viewed, it's flagged. This option allows you to delete flagged cards earlier than cards that have not been viewed yet. This way you'll keep your database small and clean. If a user comes to view their card, the card will be flagged and will have \$elapsed\_days\_two days left before it's deleted. This means that if a user wants to view their card again, he/she can do so within 2 days. Most of the time a card isn't viewed anymore, so there is no need to keep it in the database.

### **\$elapsed\_days\_urls**

Every call from a postcard desk is logged and stored in the URLs database. This database is used for the statistics and for the random postcard desk option. Because of the random postcard desk option it's very important to keep this database up to date. If a card desk didn't make a request in \$elapsed\_days\_urls days, than the desk is removed from the database.

### **\$send\_old\_cards**

1 = YES, 0 = NO. If you set this option to 1, than the script will email the card, which haven't been picked-up within the correct period (\$elapsed\_days).

### **\$send\_welcome\_email**

1 = YES, 0 = NO. If you use the script for a service you might like this option. It sends a welcome email to every new owner of a postcard desk. Inside the email people will find information about how to contact you, etc.

### **\$add\_footer**

1 = YES, 0 = NO. As a service, you need some form of advertisement. The footer in every email is ideal for this.

### **\$bgcolor**

This is the HTML code for the standard background color of a generated page.

### **\$background**

This is the URL to the standard background image of a generated page.

### **\$link\_color**

This is the HTML code for the standard link color of a generated page.

### **\$vlink\_color**

This is the HTML code for the standard visited link color of a generated page.

### **\$alink\_color**

This is the HTML code for the standard active link color of a generated page.

### **\$text\_color**

This is the HTML code for the standard text color of a generated page.

### **\$postcard\_bgcolor**

This is the HTML code for the standard postcard background color of a generated page. With postcard background we mean the background of the area where the message is located.

### **\$postcard\_text\_color**

This is the HTML code for the standard text color of the message of a generated page.

## **Advanced setup**

### **\$image\_locations**

This is the URL to the directory where you store your \$stamp\_image etc. By default this is a subdirectory of the directory where the script is located. Please check "Common Problems" if you have trouble seeing the images on the page.

### **\$stamp\_image**

This is the name of the stamp image. The image should be located in the \$image\_locations directory.

### **\$logo\_image**

This is the name of the logo image. This image is displayed at the top-middle on the backside of the card. (Internal layout)

### **\$line\_image**

This is the name of the line image. This image is simply a 1x1 pixel image, which by default is black. This image is used in the Internal layout to make a line between the message and the "TO: NAME" field on the back of the card. The image is also used in the statistics area to make the 100% line.

### **\$noimg\_image**

This is the name of the "noimage" image. This image is displayed when the getpic script has difficulties to get the correct image from the server.

### **\$pixel**

This is the name of the pixel image, which is (just like the \$line\_image) no more than a 1x1 pixel image. The only difference is that by default the image is red. It's used in the statistics area.

### **\$mtvwebdesignlib**

This should be the PATH to the mtvwebdesign.lib file.

### **\$dbdir**

This should be the path to the directory where you want to store your databases.

### **\$dbunder**

This should be the path to the dbUnder.lib file.

### **\$languagepack\_dir**

This should be the path to the directory where you store your language packs (en.lng)

### **\$terms\_dir**

This should be the path to the directory where you store your terms.txt and privacy.txt files.

### **\$maintenance\_file**

The maintenance file is an option, which you probably won't use very often, but it's ideal when you make changes to the script, etc. Every time the script is run, it'll check for this file, and if it exists, it'll read the file, print the file to the browser, and stop further processing. This variable should be the path to this file.

### **\$plugin\_dir**

This should be the path to the directory where you store your plugins.

### **@web\_email**

This is an array with all the email address which will have a webmaster effect on the script (See "Webmaster options" for more information). When using the admin.cgi script for installation you can simply enter email addresses and separate them by a comma. If you manually setup the script, you need to type this in the following format:

```
@web_email = ( 'email@email.com', 'email2@email.com' );  
etc.
```

### **\$pgen\_location**

This should be the URL to the pgen.cgi (or pgen.pl) file, which is created when installing the page generator.

### **\$getpic\_script**

This should be the URL to the getpic.cgi (or getpic.pl) file, which is create when installing the script.

### **@referrers**

If you run the script as a service, you need to leave this option empty. Else you can use this option to allow only your servers to make use of the script. When using the admin.cgi file for installation, you can simply type your servers in the text box and separate them by a comma. If you manually install the script, you need to type this in the following format:

```
@referrers = ( "www.domain.com", "name.name.com", "name.server.com" );  
etc.
```

## **Plugins Setup**

### **\$sendpics\_plugin**

This should be the path to the sendpics.pl file.

### **\$tosend\_plugin**

This should be the path to the tosend.pl file.

### **\$layout\_plugin**

This should be the path to the layout.pl file.

### **\$filestore\_plugin**

This should be the path to the filestore.pl file.

### **FileStore plugin**

#### **\$allow\_image\_storage**

1 = YES, 0 = NO. Even if you own the filestore plugin, you might don't want to use the plugin, or simply disallow the storage of images on your server. You can than set this option to 0. If you do want to allow image storage, set this option to 1.

#### **\$path\_to\_storage\_dir**

This is the path to the directory where the script will store the uploaded image. Please note that the script needs to be able to write to the directory, that's why the directory by default is a subdirectory of the plugins directory.

#### **\$url\_to\_storage\_dir**

This is the URL to the \$path\_to\_storage\_dir directory.

#### **\$max\_file\_size**

This is the maximum file size you allow, in bytes. So, if you want to allow 70Kb images, you set this option to 70000. Please note: this is not the actual allowed size of the image. It's actually the allowed size of the information stream sent to the script. For example, if a user makes use the filestore option, but types a message that is 10Kb in total, he/she has only 60Kb left for the image.

### **Layout plugin**

#### **\$path\_to\_layouts\_dir**

This is the path to the directory where you store your layouts. (See Plugins for more information).

#### **\$std\_layout**

This is the NAME of the standard layout you want to use. If the layout doesn't exist, or if this option is empty, the internal layout is used.

### **SendPics plugin**

#### **\$allow\_sending\_html\_cards**

1 = YES, 0 = NO. Even if you own the sendpics plugin, you might not want to use it. Than set this option to 0.

#### **\$path\_to\_image\_dir**

This is the PATH to the \$image\_locations directory, you configured in the advanced setup page.

#### **\$send\_old\_cards\_as\_html**

1 = YES, 0 = NO. If you have set the \$send\_old\_cards (Normal setup question 16), you might like this option. It'll try to retrieve the image and send the email as HTML.

**\$all\_as\_html**

1 = YES, 0 = NO. Do you want to send all the cards as HTML emails? Default is NO (0).

**ToSend plugin****\$allow\_specific\_date**

1 = YES, 0 = NO. Even if you own the ToSend plugin, you might not want to use the option of sending cards on a specific date. Than set this option to "0".

**\$num\_days\_adv**

To make sure your databases stay small, we have created this option. It's the number of days people are allowed to send cards in advance. This way you make sure people won't send ecards for the next 2 years or something.

## Creating a postcard desk

To create a postcard desk, simply type the URL to the pgen.cgi (or pgen.pl) file in the location bar of your browser.

You'll then see a form, which you need fill out. You can then click the submit button, and you'll be taken to a second page where you can submit the image filenames.

After you have done all this, you'll get the result. You need to store this page on your computer hard disk. (Menu File->Save as...) You might get a warning, but you can ignore that.

The next thing you need to do is upload the file to your own server, including the images used on the page, and the images needed for the postcards.

Your postcard desk is now ready!

Add more pictures after the page has been generated

If you want to add more pictures, you need to open your postcard desk in your HTML editor. If you know how to work with tables, you are lucky, because you can extend the table and add more pictures. Each picture needs a thumbnail (a small image), and a radio button, so people can select the image.

The radio button needs the following elements:

TYPE = RADIO

NAME=FILENAME

VALUE=name\_of\_large\_pic.jpg

So it'll look like:

```
<input type="radio" name="filename" value="name_of_large_pic.jpg">
```

In this case, "filename" is NOT the name of a file! This is the NAME you give the input tag, and the name is read by the script. The script reads: filename=name\_of\_large\_pic.jpg.

If you want to add more sound files, it's almost the same as the thing with the images. The only difference is that "name="filename"" should be "name="sound"".

# List of variables available in a postcard desk

Below is a list of variables you can use if you manually create a postcard desk.

## Hidden fields

(All have the following syntax: `<input type="hidden" name="VARIABLE" value="VALUE">`)

### **postcard\_email\_sender**

This is your email address.

### **language (or lng)**

This is the country code of the language you want to use for the output pages. Please note that English is "en". If not specified, the default language is used.

### **base\_url**

If specified, the script will add this URL to each filename, which doesn't begin with http://. For example, if base\_url is set to "http://www.domain.com/images/" and a person selects an image which has the value "name.jpg", the filename will become: <http://www.domain.com/images/name.jpg>.

### **postcard\_page**

This is the URL to the page where people can select images they want to send.

### **pickupurl**

This is the URL to the page where people can type their ticket number to pickup their card. If this option is not set, the script will take the page where the images are located as the pickup page.

### **sitetitle**

This is the title of your site. It's used in the email: You received a postcard from SITETITLE!

### **return\_from\_sent\_postcard**

This is the FULL URL to a page which will tell people that their card has been sent. If not specified, the script will output a page.

### **postcard\_email\_subject**

This is the subject of the email send to the person who is receiving the ecard.

### **postcard\_email\_pickedup\_subject**

This is the subject of the email send to the person who sent the postcard when the receiver viewed the card.

### **bookmark\_url**

This is the URL which is placed in the email sent to the sender when the card was received.

### **text\_color**

This can be a HTML value for the color of the text on generated pages.



**bgcolor**

This can be a HTML value for the background color of generated pages.

**link\_color**

This can be a HTML value for the link color on generated pages.

**vlink\_color**

This can be a HTML value for the visited link color on generated pages.

**alink\_color**

This can be a HTML value for the active link color on generated pages.

**background**

This can be the URL to a background image for the generated pages.

**postcard\_bgcolor**

This can be a HTML value for the background color of the backside of the postcard on generated pages.

**postcard\_text\_color**

This can be a HTML value for the text color of the backside of the postcard on generated pages.

**cardclick**

1 = YES, 0 = NO. If you use this option, and link will be displayed in the email of the receiver so that he/she can simply click that link to view their card.

**layout**

If you have different layouts available you can set the NAME of the layout here. Please see "Plugins" for more information.

**font\_face**

This is the face of the font which will be used on generated pages. E.g.: Arial, Helvetica, etc.

## **User defined fields**

**filename**

This is a set of radio buttons so people can choose an image. The value of this variable should match a filename on your server. If the filename doesn't begin with http://, the script will presume that the file is located in the BASE\_URL.

**user\_url**

This is a text field where users can enter an URL to an image somewhere on the internet.

**upload**

If you own the FileStore plugin, and you have \$allow\_image\_storage set to "1" (YES), than you can add this field. It should be a FILE input tag. Also, you have to make sure that the FORM tag has the correct ENCTYPE: multipart/form-data.

### **sound**

This is a set of radio buttons so people can choose a sound file. The value of this variable should match a filename on your server. If the filename doesn't begin with http://, the script will presume that the file is located in the BASE\_URL.

### **text**

This is a TEXTAREA where people can type their message.

### **sender**

This is a text field where the sender can enter his/her name.

### **senderemail**

This is a text field where the sender can enter his/her email address.

### **recipient**

This is a text field where the sender can enter the name of the recipient.

### **email**

This is a text field where the sender can enter the email address of the recipient.

### **month**

This is a list of months. January = 0, December = 11. This option is only available if you own the ToSend plugin.

### **day**

This is a list of 31 days.

### **year**

This is a list with years.

### **action**

This is a set of radio buttons, so people can choose if they want to preview the card first, or email the card immediately. You can also use other types of course, like a select list or something. There are 2 options available on the postcard desk: send and preview.

You need to create a separate form for receiving the card. In this form, you need to include an action too, but this time, you'll need to make a hidden input tag: <input type="hidden" name="action" value="receive">

### **ticket**

When receiving the card, only one user field is necessary. This is a text field where people can enter their ticket number.

Please note that you have to include the same configuration in both forms (bgcolor, text\_color, etc.), because the script will not remember all your options.

All forms need to end with a **SUBMIT** input tag, so the user can submit the form.

# Plugin information

## Installation

To install plugins, simply upload them to your plugins directory in ASCII mode, and run the admin.cgi (or admin.pl). Select the option "Install Plugins" and the script will automatically detect which plugins are available.

## FileStore plugin

The FileStore plugin enables users of your script to upload their own images to your server and so create their own ecards!

## Layout Plugin

This plugin allows you to create multiple layouts for the generated pages. To create a layout, you need to create 2 html files. One is the preview page, and the other the receive page. If you create the 2 layouts, rename the files to NAME\_preview.html and NAME\_receive.html, where NAME is the same. Make sure the files have an HTML extension, and not a HTM extension. The following variables are available for you in the 2 pages:

**%%%HEAD%%%**

This will print either "This is what the card will look like" (preview), or "Your postcard" (receive), also in different languages.

**%%%BODYTAG%%%**

This will print the body tag (and the colors specified in the postcard desk)

**%%%TERMS%%%**

This will print a link to the Terms file.

**%%%PRIVACY%%%**

This will print a link to the Privacy file.

**%%%BROWSERS%%%**

This will print a link to the section which explains which browser can be used

**%%%RANDOM%%%**

This will print a link to a random postcard desk

**%%%IMAGE%%%**

This will either print the IMG tag, or html tags for Flash files.

**%%%SOUND%%%**

This will print the tags needed for the sound file.

**%%%TURNIMAGE%%%**

This will print the line: Click here to see the back of the card.

**%%%BACKBUTTON%%%**

This will print a backbutton.

**%%%MAILBUTTON%%%**

This will print a "Mail Card" button, need in the preview page.

**%%%POSTCARD\_TEXT%%%**

This will print the postcard text.

**%%%TO%%%**

This will print "To:" in the language needed.

**%%%RECIPIENT\_NAME%%%**

This will print the name of the recipient.

**%%%RECIPIENT\_EMAIL%%%**

This will print the email address of the recipient.

**%%%SENDER\_NAME%%%**

This will print the name of the sender.

**%%%SENDER\_EMAIL%%%**

This will print the email address of the sender.

**%%%SIGN\_OUT\_TEXT%%%**

This will print the sign-out-text: Click here to return to the postcard desk to send....

**%%%POSTCARD\_TEXT\_COLOR%%%**

This will print the postcard\_text\_color in HTML values

**%%%POSTCARD\_BGCOLOR%%%**

This will print the postcard background color in HTML values.

**%%%PAGE\_URL%%%**

This will print the URL to the page the card was sent from.

**%%%SITETITLE%%%**

This will print the title of the site the name was sent from.

**%%%STAMP%%%**

This will print the IMG tag for the stamp image.

**%%%LOGO%%%**

This will print the IMG tag for the logo image

**%%%LINEIMAGE%%%**

This will print the IMG tag for the black pixel, which will become a line.

A small example: If you want to create a link back to the site the card was sent from, you can use:

```
<a href="%%%PAGE_URL%%%">%%%SITETITLE%%%</a>
```

## **SendPics Plugin**

This plugin enables you to send HTML emails. So the recipient will not receive an email with a ticket number, but he/she will receive an email with the image inside! The SendPics plugin now also uses the correct layout, and not the standard internal layout.

## **ToSend plugin**

This plugin enables users to send cards on a specific date.

## Webmaster options

Most webmaster options are available in the admin.cgi script. But there are some hidden options.

If you use an email address which you specified as a web\_email for sending a card, you'll never be redirected to a return\_from\_sent\_postcard page, but you'll get a special page which will have the ticket number and a link to the correct page (if it exists). This way you'll never have to wait for the email to arrive to receive the card.

When people complain that they cannot see a card, you can check if it exists using a special webmaster option. In the ticket field, type the ticket number and add ".web". The card will be displayed, but the email which tells the user that the card has been picked up will not be sent, and the card will not be flagged for deletion. For example, if you have the following ticket number:

aa12345

Your webmaster ticket number will be:

aa12345.web

# Common Problems and solutions

## Languagepacks

You might have problems when installing new language packs. It's advised that when a person comes to you and asks for his/her own language, you send the English Language pack, en.lng. Inside this file, which should be opened in an ASCII based editor like NotePad or SimplePad, are instructions on how to use the file.

When the person returns the file to you, you simply need to upload the file to the language pack directory for it to become active.

When you get an Internal Server error when using the script with the new language pack, you should check the following things:

Did you upload in ASCII?

Are there any quotes (") in edited lines. If so, remove them, you should only have quotes at the beginning and end of a line of text.

### How to get the language pack active in the page generator.

Open the admin.cgi (or admin.pl) file and find the hash called "languages", which you can find around the beginning of the file and looks something like:

```
%languages = (  
  "English" => "en",  
  "Dutch" => "nl",  
);
```

To install the new language pack, simply copy the "English" line, and paste it one line down. Then type the new language over the English, store the file, upload the new admin.cgi file in ASCII mode to your server, and try to run it. Then re-install the page generator and you're done!



## Matching site layout with page generator

The page generator has an option so that the page generator question form (the first to steps of the page generator) will have the same look-and-feel as your site. This is done with the use of a tokenfile. A tokenfile is nothing more than one of your pages. The only thing is that the page doesn't have any text yet. Instead of the text it has a token, an HTML comment.

By default the script doesn't use a tokenfile, but you can set question 3 in the page generator installation form to YES if you do want to use a tokenfile.

The default tokenfile is `.tokenfile.html`, and the default token is: `<!-- TOKEN -->`.

### How to create a tokenfile

Let's say you have the following page as one of your webpages on your site:

```
<html>
  <head>
    <title>My Webpage</title>
  </head>
  <body background="image.gif">
    <center><FONT size="+1">Welcome!</font></center>
    Hello there, welcome to my page!
    <hr>
    <center>
      Contact information:<br>
      NAME, email: <a href="mailto:email@email.com">email@email.com</a>
    </center>
  </body>
</html>
```

Now, you want the page generator question form to have the same background image, and the contact information, but you do not need the big welcome, and you don't need the Hello there, welcome to my page.

What you need to do is simply delete the lines you don't want. Then you determine where you want the script to place its output, and there you add the token, which is `<!-- TOKEN -->` by default, so we'll use that here. The result is this:

```
<html>
  <head>
    <title>My Webpage</title>
  </head>
  <body background="image.gif">
    <!-- TOKEN -->
    <hr>
    <center>
      Contact information:<br>
      NAME, email: <a href="mailto:email@email.com">email@email.com</a>
    </center>
  </body>
</html>
```

The page generator will now have the same look as all your other pages.

Please note that the TOKEN has to be an HTML comment, and it's advised to place alone on one line to get the best results.

The last thing you need to do is upload the file to the location you specified in question 4 of the page generator installation form. By default the name is `tokenfile.html` and is located in the same directory as the `pgen.cgi` (or `pgen.pl`) file.