

**How CGI Works**  
and  
**Answers to Other Questions  
You Didn't Know  
You Could Ask**

by  
William Bontrager



**A Peach-e-Book**  
Copyright 2001 Bontrager Connection, LLC

## Introduction

You hold in your virtual hands a compilation of several articles previously published in **WillMaster Possibilities**, a free weekly e-zine from **Bontrager Connection, LLC**.

From time to time in these chapters, you will find mention of the **Master Series** - the line of CGI software offered by **WillMaster.com**. We think you will be pleased to discover that many of the software titles in the **Master Series** are free and, as such, are an excellent introduction into the world of CGI and automating your web site.

All material in this publication are © 2001 Bontrager Connection, LLC. All rights reserved.

This entire ebook, without alteration, may be freely distributed.

Reprint rights for specific articles may be obtained at **WillMaster Possibilities**.

If the Bontrager team can be of any further assistance to you, please feel free to **contact us**.

## **How CGI Works**

When you understand this article, you will have

- an intuitive understanding of what CGI is.
- a better grasp of what is possible and what is not.
- more confidence when installing CGI programs.

You already understand that when you surf the internet your browser retrieves (or gets served) information from other computers.

Those other computers send (or serve) information to your browser with special computer software called a server. Every computer that serves information to internet browsers uses server software.

Your browser asks the server software for something and the server software gets it for you.

You can compare it to a restaurant. You give your order to the server (the wait person) and the server gets it for you.

Just as you always have to go through the restaurant's server person to get food, so your browser always has to go through a server program to get information.

Most restaurants have food ready to be served immediately, such as drinks and snacks, and also have food that requires the skills of a cook to prepare before serving.

Most remote computers on the internet have information ready to be served immediately, such as web pages and images, and also have information that requires processing before serving.

When you want a custom prepared meal, you order it through your server person. The server takes it to another person. This person is called a "cook." The cook prepares the meal by gathering together the appropriate ingredients and combining them correctly; then gives it to your server. Your server delivers the meal to you.

When your browser wants custom information, the browser tells the remote computer's server software. The server hands it to another program on that computer. That program is called a "CGI program". The CGI program gathers together the appropriate data and does any required calculations on it; then gives it to the server. The server delivers the information to your browser.

The custom prepared meals you can get at a restaurant are limited to the kitchen equipment at hand, the skills of the cook, and the available ingredients.

The custom information you can get with your browser is limited to the operating system of the remote computer, the type and quality of CGI programs on the computer, and the available data.

The servers at a restaurant have a specific method of sending requests to the cooks and receiving meals from the cooks. That method could be called the server/cook interface; it is a common gateway through which requests are made and meals received.

The server software on a remote computer has to follow a specific method (called a "protocol") to send requests to CGI programs and receive custom information back. That protocol is a server/program interface. It is a common software gateway through which requests are made and custom information received. It is called "Common Gateway Interface" or "CGI".

Let's suppose you order a meal in a restaurant and the server person sends your order to a cook. The meal is taking a long time and you have another appointment. So you leave the restaurant before your meal arrives.

The server person at the restaurant notices you're gone and tells the cook. So the cook stops cooking your meal -- no sense putting any more effort into it when you're not there to receive your order.

Let's suppose you're at a web page and click a link. The remote computer's server software sends the request to a CGI program for fulfillment. It's taking a long time and you're in a

hurry. So you leave the site and go surf somewhere else.

The server software on the remote computer notices your browser is gone and tells the CGI program. The CGI program quits running -- no sense using computer resources when you're not there to receive the information.

That's how CGI works.

Computer programs are CGI programs **only** when they communicate with the server using the CGI protocol.

Computer programs (including CGI programs) can be written with any programming language that the computer's operating system can understand. Perl is currently the most popular programming language used for CGI programs, but it is definitely not the only one so used.

The limitations of CGI are limited only by the browser you're using, the remote computer's limitations, and the programs the remote computer has on board.

So, how about a real-world example?

One of the most broadly useful CGI scripts is one that processes data from a feedback/contact web page form.

The example uses our own Master Feedback, available at <http://willmaster.com/master/feedback/index.shtml> .

A web page is in your browser. The page contains a form.

You fill in the form and click the submit button.

The browser copies the information you provided on the form, along with the name of the CGI program that will process the information (in this case, Master Feedback).

The browser sends the information to the remote computer's server. (Your browser doesn't make decisions about what to do with the information. Everything just gets sent to the server. The server then makes the decisions.)

The server looks at the information your browser sent and decides what to do with it. In this case, it makes the information available to Master Feedback.

In between the time when your browser sends the information to the server and the time it hears back from the server, Master Feedback:

1. Accepts the information from the server.
2. Mails the form contents to the appropriate email address.
3. Gives the server the URL of a "thank you" page.

The server accepts the URL from Master Form and gives it to your browser. Your browser goes to that URL -- which means the next page you see is a "thank you" page.

That's the complete CGI cycle!

## **The Magic of CGI!**

Without CGI, there would be no contact forms on web sites, no live surveys, no shopping carts, no search engines. Web sites would be static things that provide information and pretty pictures but offer no interaction.

Imagine this:

Someone in Germany can surf to your web site (which might be located on a computer at a hosting company in Ontario, Canada) and answer a quiz on one of your pages in exchange for a discount to your membership site.

The credit card purchase might go through a payment gateway on an internet server located in Chicago, USA. After the customer's credit card is approved, the customer fills in a form on your membership site's domain (which might be at a different hosting company, anywhere in the world) with preferred username, password, and other information -- and gains instant access as a member.

The quiz results, the customer's contact information, the discount earned, the purchase particulars, and the new member's username and other information are automatically emailed to your mailbox at your ISP, which may be in the city of Sydney, Australia. All of that activity takes place in your absence while you're enjoying a bite of dessert after a particularly satisfying meal.

A few minutes later, you download your email and say, "Oh, good. Another sale." And you use the forum in your membership area to personally welcome the new member.

Without CGI, that magical scenario simply wouldn't happen.

More than CGI is involved, of course. Without the magic of the internet itself, CGI would be useless.

And, yes, other technologies can do things CGI can do. They tend to specialize. Some do their job quite well. But CGI was first, it is still the best in its class, and it remains the most popular. It is a proven technology that works with all browsers and requires no plug-ins.

CGI is a method of receiving information provided by site visitors through their browsers. It's instant. (Well, it may take a moment or two to process the data.) CGI is also a method of sending information to visitors, again via their browsers. Information is received and responded to automatically. It's two-way communication. It's live. No webmaster intervention is required.

CGI cooperates well with other technologies, too. For example, our content syndication programs at the **Master Series** (**Master Syndicator** and **Master Syndication Gateway I**) use CGI to receive content and prepare it for syndication. Then, JavaScript is used to deliver the content to remote syndication sites.

The webmasters of your remote syndication sites only need to paste a couple lines of JavaScript into one web page, at the spot where they want your content to show up. That's all they have to do; ever. (Well, they'll need to pay you every once in a while, if you charge for your content.)

When you put new content into the special form on your site and click the submit button, all remote syndication sites with the special JavaScript code on their pages are updated instantly and automatically.

Pure magic!

With CGI, you can have

- Shopping carts
- Forms
- Database driven sites
- Visitor tracking with real-time reporting
- Interactive server maintenance
- Information retrieved from remote web pages
- Databases updated on remote servers
- And so much more

Take a database-driven site, for example. Every page your visitor views is freshly generated, live on-the-spot, from your databases -- products, prices, contact information, articles, bulletin board threads, everything. When you want to update information, you probably use

a form in your browser. The moment you click the submit button, the new information is already available to your visitors.

Yes, just like magic.

With CGI, you can

- Allow your visitors to **recommend your web site**.
- Let your visitors **request information selected from a check list**; it's emailed instantly.
- Automatically **rotate content** on your site.
- Run a **forum**.
- Keep a **weblog**.
- Take a **survey** or **poll**.
- Offer a **quiz**.
- **Add new website content**, almost automatically.
- Provide a **search engine**.
- Manage your **reciprocal links** page.
- **Track your links** for your newsletters or ad campaigns.
- Keep a password protected "**Members Only**" website.
- View the **source code** of gateway pages, framesets, other pages normally cloaked by browsers.

You get the idea. Works like magic!

## **Be a Good Host; Let Your Visitors Speak**

How hard is it for your site visitors to send you a message? Can they do it at all?

At a business site, your visitors sustain you.

Not providing contact methods is rude; it can result in missed opportunities to improve your site and raise your income. Not listening to what your visitors have to say is impolite.

Here are a few examples of how you and your business might benefit:

- A visitor tells you about something you could easily implement for better site navigation.
- There is an obvious and embarrassing typographical or grammatical error on a page, and someone lets you know.
- You are notified that a download link consistently retrieves a broken file.

- Someone makes a request that gives you an idea that will result in your financial independence.

- You can be told about problems with your product or service, rather than have the customer spread dissatisfaction noises throughout the internet with "... and their web site doesn't even say how to contact them!"

- A prospect may need more information about a product before they will buy. If they can't get it, they aren't gonna :)

Without feedback, site owners can't respond to what their visitors want and expect or what tests their patience. To stay in business, your site must be responsive.

Besides, not providing contact methods is disrespectful of your visitor's needs, intelligence, and good will.

For most sites, contact methods should include all of the following:

- Email address (see below about spammer's email address harvesters).
- Postal mailing address, including country name.
- Telephone number.
- A feedback/contact form.

Give your visitor a choice and s/he will be much happier.

If some contact information is absent, visitors may get the impression you are trying to hide. That will destroy the customer confidence your business requires.

Specifying an email address can be fodder for spammer's email harvesting robots. To prevent your address from being harvested in that way, create a graphic with your email address on it. Then use the image instead of text.

If you link the image to an href="mailto:\_\_\_\_\_" link, the email harvesting robots can grab it. See "Advanced Email Link Generator with Anti-Spam Encoder" at <http://willmaster.com/possibilities/demo/aelgwase.html> for some protection.

For best spammer protection, omit all email addresses from your source code. We use the image method, which allows only people to have access to our email address, and a contact form. Most people prefer to use the contact form.

The free CGI script, Master Feedback, from <http://willmaster.com/master/feedback/> is for simple contact forms. Master Feedback is designed to help you spam-proof your site.

The commercial CGI script, Master Form, from <http://willmaster.com/master/form/> has more features, including the ability to store collected information in a database. Master Form could be useful if your contact or other forms need to be more complex.

If you do not have CGI available for your site, there are sites that will host your scripts. Some charge a fee. Others do it for free but display advertising. The <http://feedback.com/>

site provides a free form service with ads.

You need a contact form. It shows respect for your visitors by providing an easy way to communicate with you. A form is easier than an email link because the form is right there and an email link needs to launch the email program before your visitor can begin typing.

Responding to your visitors in a timely manner with appropriate answers can generate much goodwill and loyal customers.

I've seen the argument saying email links are preferable to contact forms because it ensures a valid email address. But when you think it through, that reasoning has several flaws.

First, and of least consequence, most people know how to type their email address and rarely make intentional typos. Yes, there are a few, and if that bothers you then you may want to forego the contact form. But if you use either of the CGI programs mentioned above, they will catch email addresses that are incorrectly formatted.

Second, using a contact form allows your visitors to send you anonymous email by providing a fake, albeit correctly formatted, email address. This is important. Some people will say constructive things about your site so long as they know you won't bother them with email they don't want. Many people feel their email addresses are being abused and don't like to give them out. Also, with anonymous messages people tend to be more honest about their feelings.

Third, a form is so easy that some people will use it who wouldn't otherwise bother.

An additional option for visitor feedback is using a live customer service program. Sites that can stimulate more spontaneous purchases by having someone on hand to answer questions would benefit. <http://humanclick.com/> is one site offering such a solution.

Some sites, however, might find live customer service a distraction or a commitment they are not yet ready to assume. Feedback, answered promptly, is sufficient responsiveness for most sites.

However you do it, be a good host. Let your visitors speak.

## **Be a Good Host; Make it Easy to Recommend your Site**

The best visitor for your site is one who is interested in your subject matter.

One way to get interested visitors to your site is to let your current visitors bring them to you. When someone likes your site and voluntarily recommends it to a friend, that friend is almost certain to be interested in what you have.

And those that are interested in what you have are more likely to recommend your site to their own friends.

It stands to reason, then: make it easy to recommend your site.

Yet, many webmasters don't do it. Some hide the recommend link deep down the page. Some insert it among a confusion of links. Some don't have a link at all.

### **Requirements**

There are two requirements for a successful "recommend me" implementation:

1. The site must be something visitors want to recommend.

If the site is not something a person would want to recommend to their friends, then product or monetary incentives probably won't work either. Unless recommendations are voluntary, quality of recommendations become suspect because the visitor is prone to make recommendations for self-gratification rather than doing a favor for a friend.

## 2. Making a recommendation must be easy.

If the visitor has to hunt for a recommend link, then it's not easy.

Put a prominent recommend link (not imploring, just available) at every point on your site where a visitor might think, "Oh, my friend would want to know about this." If the visitor has to hunt for a link, the urge to recommend your site is likely to dissipate -- you lose a potential sale and possibly additional recommendations that could convert to sales.

## **Abuse Potential**

In our experience abuse is rare. We've had two instances in the many years we've been using recommendation forms on our sites.

One was the use of a really nasty term that got by our offensive language filters. We added that term to the list right away! The other was by someone who was harassing a specific individual but carefully avoiding words that might trip the filters. We blocked the destination email address.

Although rare, if abuse does happen then preventative steps need to be taken right away.

## **Implementation**

There are two ways to implement "recommend me" functionality:

1. Using a third-party service
2. Putting the software on your server.

Third-party services are the easiest to implement. Often, it is as simple as signing up and getting a unique link to put on your site. The drawback is that you don't control the mail and, especially if the service is free, your visitors and their friends will be subject to other people's advertisements. Probably the most popular third-party service of its kind is [recommend-it.com](http://recommend-it.com).

If your server has its own cgi-bin, I recommend having your own implementation. Install the script and be in control.

If your server is UNIX or Linux, I recommend our own programs:

1. Master Recommend from <http://willmaster.com/master/recommend/> has been downloaded about 10,000 times since it was released on April 13, 1998.

It's still free and probably will remain free.

Master Recommend has a offensive language filter, customizable outgoing emails, webmaster notification, and a message area for the sender's comments. It supports only one recommendation recipient. The month and weekday names may be in any written language. Visitor gets a thank you note and can also elect to receive a copy of what's sent to friends.

The script may be modified to alter the list of offensive words/terms. It may also be modified to refuse to send mail if the recipient is a banned email address.

2. Master Recommend Pro from <http://willmaster.com/master/recommendpro/> is a commercial program for \$45 (USD).

It features a control panel where you generate recommend forms with any number of recipients you please, list banned addresses, update your list of "offensive" words/phrases, specify which domains the program will recognize as valid, maintain email templates for custom outgoing email, and more. It supports dates and mail in any written language. Dates may be formatted any way you like. Form information can be logged. "Thank you" pages may be personalized, including another recommend form partially pre-filled in. Visitor can get a thank you note and can also elect to receive a copy of what's sent to friends.

Master Recommend Pro is designed to be very easy for the webmaster to use and administer.

Master Recommend Pro is also visitor friendly in a way few other scripts are. If the visitor

reverses the information for the name and email fields, the program will switch them automatically.

The main differences between the Master Recommend and Master Recommend Pro programs is:

**Master Recommend** requires modifying the script itself for configuration and changes. Only one recipient can be specified at a time. The form is created by hand.

**Master Recommend Pro** has a control panel for all configurations, formatting, email templates, etc. Any number of recipients can be specified in the form. The control panel generates your forms so all you need to do is copy and paste into your pages.

Master Recommend and Master Recommend Pro require UNIX/Linux servers. If your server is NT, then the free BirdCast from <http://www.bignosebird.com/carchive/birdcast.shtml> is popular. Because BirdCast is not our program, I am not intimately familiar with it. Their site states that BirdCast has unlimited number of recipients, logging of pages referred, and a message area for the sender's comments.

Your site deserves all the recognition it gets. Be a good host and make it easy for your visitors to recommend it to their friends.

## Why CGI?

With alternatives available, why CGI?

Ah, glad you asked!

Number one reason is that CGI is a technology proven to work with every browser and all major server operating systems.

CGI does not make browsers crash. And CGI requires no special plug-ins.

That means you don't have to worry about whether or not it can work. Of course it can work. If it doesn't, it's not the CGI protocol at fault. Rather, it would be the CGI program or its installation. And because of CGI's popularity, it's easy to find answers or get professional help when a program doesn't work or the installation is amiss.

Let's back up a step and mention that CGI is not a programming language.

Rather, it is a method or protocol, something that serves as an interface or gateway between the browser and a program on the server. The browser asks for information and the program provides it. The program itself can be written with Perl, C++, or most any other language the server understands.

With CGI, you can have shopping carts, forms, database driven sites, visitor tracking, interactive server maintenance, info retrieved from remote web pages, databases updated on remote servers, and so much more. Peruse CGI script download sites to gain a preliminary understanding of what's available for use with CGI.

It's CGI that makes most interactive, automated sites work as smoothly as they do.

Alternate technologies, such as ASP, PHP, and ColdFusion, do some of the same things CGI does. Some alternates work better on one server and might not work on others. ASP and ColdFusion were developed for NT, for example. Both now work on some UNIX servers, but not on all. PHP is popular from sheer force of merit. It is relatively easy to learn and is excellent with database query languages such as MySQL.

However, these are newcomers compared to CGI. And none of them have as many free and complete scripts available for immediate download.

CGI solutions are more available for UNIX/Linux servers than for NT. If you're looking for a hosting account with CGI, that can be a major consideration.

Without CGI, you must either depend on alternate technologies or forego much of your site's potential for user interactivity and automation.

## **Shopping for a Hosting Account**

Your hosting account must allow you to install your own CGI programs. Ensure that you not only have CGI access but that you have complete control over your own cgi-bin. (It is called "cgi-bin" because, traditionally, CGI programs reside in a directory named "cgi-bin" -- however, the directory can be named anything allowable by the server.)

Mere CGI access can mean, to some hosting companies, that you can use their authorized programs in the cgi-bin but you may not install your own without prior permission. Ensure your account has its own cgi-bin and that you have complete control over it.

Because Perl is the most popular language used for CGI programs, another consideration is the availability of Perl modules. A Perl module is a piece of program code designed to do specific functions and is installed on a server for use by all of that server's hosting accounts.

For example, LWP::Simple is a popular module that allows Perl programs to retrieve web pages from remote servers. Your hosting company should have that installed and ready for your Perl program's use.

Most hosting companies are happy to install popular Perl modules when their accounts require them. New Perl modules are written all the time and no hosting company can be expected to have all of them installed as soon as they are released. However, the current popular modules should already be installed and the hosting company should be willing to install new modules as they become available and their hosting accounts ask for them.

Once you have a server with your own cgi-bin, you can use the free Master Pre-Installation Tester from <http://willmaster.com/master/pit/> to check whether or not specific modules are installed on your server. Some Perl CGI programs require certain modules, without which they simply won't run.

WillMaster Possibilities free weekly e-zine delivers new ideas to your email box every Tuesday. Isn't it time you were a subscriber? [Click here and begin.](#)