



Web Site Mini-Guide: Forms and Surveys

You can use this to create feedback forms on your Web site.

Overview.....	2
Planning Your Form	2
Creating the form	4
Field Types.....	9
Layout guidelines	15
Error Page.....	18
Follow-Up Page	19
Testing Your Form	20
Downloading Results.....	21
Other Options	23
Using Other Web Design Software.....	24
More Help	26



You can create feedback forms on your Web site. You can use them for various things, such as:

- Gathering names and addresses of site visitors
- Conducting surveys of site visitors
- Asking site visitors for information before they download something from your site

This guide describes how to plan your form, create it and test it.

Overview

You create the form as an HTML page on your Web site, in the same way as you create other HTML pages.

The form is “linked” to the E-Service Form module, which takes the user input, validates it, e-mails it to you, and saves a copy on the server.

It then displays a “follow up” page, which is also on your Web site.

You can create as many forms as you like on your Web site.

Planning Your Form

Put some thought into planning your forms so that they are easy as possible for people to use.

Feedback Forms

If you're asking people on your Web site to give you feedback, keep these things in mind:



- Keep the forms as simple as possible. Research shows that the less information you ask for, the more responses you will get.
- Present the fields in a logical order, and group related fields together. For example, if you're asking for contact information such as phone numbers and e-mail address, put all these fields in the same area of the form.
- Decide which fields are compulsory and which are optional. Make as many of them optional as possible, unless it's essential that you have this information.
- **Note:** It's useful to have at least one compulsory field on the form. This forces the user to fill in at least one option, and avoids the situation where a user comes across your page and idly submits your form just to see what happens.
- When asking for contact details, be aware of international users. For example, don't present a drop-down list for State that only allows US States! And don't make State a compulsory field, because some countries don't have States (Singapore, for example).
- If you're asking the user for personal information, tell them how you plan to use it. For example, you could include a prominent link to your privacy policy.
- If you're going to provide a list of options, using a drop-down list makes it easier for the user than asking them to type a response. But make sure that you do include all possible options. If the user could type in some other option that's not in your list, add "Other" as an option in the list and create a separate field for them to type in their choice.

Surveys

Many of the guidelines above apply equally to survey forms. In addition, surveys have their own set of requirements as well, including:



- Tell the user the purpose of the survey.
- Give clear instructions on how they should complete the survey.
- If it's a detailed survey, tell them approximately how much time to set aside.
- Tell them how you will use the survey results. This is especially important if the survey identifies them personally, and they are worried about confidentiality.
- Be careful of long questions, confusing questions, questions with double or triple negatives ("Do you disagree with the boycott on companies who ban the use of X in the workplace?") and loaded questions.
- If you have a multiple-choice question **and** it's compulsory, a "None of the above" option is often appropriate (It might be called "Not applicable", "Not sure", "I don't know" or similar wording).
- Add a "Comments" field at the end for the user to provide any other comments that don't fit in the rest of the survey.
- If you don't **have** to identify the user, make the survey anonymous. You will get more honest feedback, especially if it's a sensitive topic.

Creating the form

This section describes the technical information for creating a form. We'll describe the basics in this section, and then give you some further layout guidelines in the "Layout Guidelines" section.

This section assumes that you're using a Fast Forward Web site designed by First Step Communications, so you will be creating your form using the "Smart Text" editing system on the



site. If you're not creating your form this way, look at the "Using Other Web Design Software" section at the end of this document.

We will demonstrate how to create a form by creating a very basic form that looks like this:

Introductory text goes here ...

Fields marked * are required; others are optional.

First Name *

Last Name *

E-mail Address

Comments:

Send Query

Note: If you already have another form on your Web site, it might be easier to copy that rather than starting from scratch.

Basic Template

Start by copying this form code, which contains the technical code and a few sample fields:

Introductory text goes here ...

```
<form action="http://fsplugins.com/cgi-bin/sendmail.pl" method=post>
<input type=hidden name="account" value="aaa">
<input type=hidden name="mail_from" value="from@example.com">
```



```

<input type=hidden name="mail_to" value="to@example.com">
<input type=hidden name="mail_subject" value="Booking Enquiry">
<input type=hidden name="followup_url" value="bookings-thanks.html">
<input type=hidden name="export" value="Booking Enquiry">

```

Fields marked `*` are required; others are optional.

```

<table border=0>
<tr><th align=left>First Name <font color="#ff0000">*</font></th>
  <td align=left><input name="First Name" type=text size=30 required>
</td></tr>
<tr><th align=left>Last Name <font color="#ff0000">*</font></th>
  <td align=left><input name="Last Name" type=text size=30 required>
</td></tr>
<tr><th align=left>E-mail Address</th>
  <td align=left><input name="Email" type=text size=30>
</td></tr>
<tr><td colspan=2><b>Comments:</b>
<br><textarea name="Comments" rows=5 cols=50></textarea>
</td>
</tr>
<tr><td colspan=2 align=center>
  <input type=submit value="Send Query" name="send">
</td></tr></table>
</form>

```

Paste this code into a Web page, and then follow the guidelines in the rest of this section to edit it to suit your purposes.

Header fields

The form begins with this line, which you should not change:



```
<form action="http://fsplugins.com/cgi-bin/sendmail.pl" method=post>
```

Note: Depending on where your site is installed, you might have to change fsplugins.com to www.eservice.com.au or fsplugins.net. If you're not sure, please check with First Step.

Change the hidden input fields directly below that line (shown in bold in the example above):

- **account:** "aaa" is your E-Service account name
- **mail_from:** "from@example.com" is the e-mail address used as the sender of the e-mail that's sent to you. This could be your own e-mail address or some other dummy address at your domain. If you ask the user for their own e-mail address, put it in a field named "e-mail", and it will override this field.
- **mail_to:** "to@example.com" is the recipient's e-mail address (normally your own). If you would like it to go to more than one e-mail address, list them here separated by commas.
- **mail_subject:** "Booking Enquiry" is the subject of the mail message
- **followup_url:** The Web page address of your follow-up page (which you will create later).
- **export:** A unique name for storing the form results on your server. Use a different name for each form so the results for each form are stored in different files. It's common to use the same name as the e-mail subject. However, if it's a long subject, use a shorter name here.

Input fields

You will then see the form fields, which in this example are First Name, Last Name, Email and Comments.

Choose meaningful names for each field name; these are sent back to you, so you have to know what they refer to.



Start each field name with a capital letter (e.g. "First Name" or "First name", **not** "first name").

Use letters, digits, spaces and dashes only in field names.

The field names must be unique. If you use any duplicate names, the form won't work.

The next section "Field Types" describes how to ask for certain types of input, such as text input, radio buttons, checkboxes and drop-down lists.

Required fields

If there are any fields that are compulsory, add the word "required" in the line for that field – e.g.

```
<input type=text name="Blah" required>
```

In the sample above, the First Name and Last Name are required fields. You can also see that there's a visual clue (a red asterisk) next to the required fields.

Submit Button

Use a line like this to create the submit button:

```
<input type=submit value="Send Enquiry" name=send>
```

You can change the text "Send Enquiry" to whatever text you would like to appear on the button.

End of Form

The form ends with this line, which you should leave exactly as it is:

```
</form>
```



Field Types

This section describes the various input field types you can use in your form.

Text

The simplest field type – and probably the most common – is a basic one-line text field – e.g. the First Name field in the sample above:

First Name *

The code for that uses the input field type **text**, like this:

```
<input name="First Name" type=
```

text

```
>
```

Block of Text

If you require more than just one line of text, you can have a block of text, like the Comments field above:

Comments:

The code for this is the field type **textarea**, like this:

```
<textarea name="Comments" rows=5 cols=50></
```

textarea

```
>
```

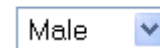


This gives a block that is 5 lines (rows=5) deep and 50 characters (cols=50) across. You can change these two numbers as required.

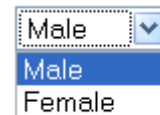
Note that these numbers only determine the **visible** part of the block. If the user types for more than 5 rows, the text block will automatically accommodate that with scroll bars.

Drop-Down Menu

Drop-down menus are common on forms. The user first sees a single option:



and then clicks the little arrow to see all available options:



Here is the code for this particular menu:

```
<select name="Sex">  
<option>Male  
<option>Female  
</select>
```

No default

You can see in this example that the first option (Male) becomes the default option in the drop-down menu. You might prefer to have no option selected by default, like this:





In this case, add a blank option at the start of the list, like this:

```
<select name="Sex">  
  <option>  
  <option>Male  
  <option>Female  
</select>
```

Different default

Or, you might prefer a different option to be the default, even if it's not the first in the list. In this case, add the word "selected" to that option, like this:

```
<select name="Sex">  
  <option>Male  
  <option selected>Female  
</select>
```

Different text

In all these cases, the text passed back to you is the text on the option the user chose (e.g. If they choose "Male", the word "Male" is sent to you). However, sometimes it's useful to display one set of options, but actually send back something different to you.

For example, you might like "M" and "F" sent back to you, but you would still like to show "Male" and "Female". Here's how to do that:

```
<select name="Sex">  
  <option value="M">Male
```



```
<option value="F">Female  
</select>
```

Radio Buttons

Radio buttons are an alternative to drop-down menus. For example, instead of the drop-down menu above, we could have:

Male
 Female

Each option is known as a radio button, and the entire thing is known as a "set" of radio buttons.

This is the code for this set of radio buttons (excluding the title):

```
<input type=radio name="Sex" value="Male"> Male  
<br><input type=radio name="Sex" value="Female"> Female
```

The important thing in a set is that every radio button has the same name (e.g. name="Sex" in this example), but has a different value (value="Male", value="Female", etc.).

Note the
 in front of the second radio button, which puts it on the next line. This would also be used for any subsequent buttons.

Specifying a default

Unlike drop-down menus, there is no default value (i.e. both Male and Female are unselected). If you would like a default value, like this:



- Male
 Female

use the word “selected” in that particular button, like this:

```
<br><input type=radio name="Sex" value="Female" selected> Female
```

Allowing it to be optional

If you would like to make this field optional, add a separate option like this:

- Male
 Female
 Not specified

Here is the code for that:

```
<input type=radio name="Sex" value="Male"> Male  
<br><input type=radio name="Sex" value="Female"> Female  
<br><input type=radio name="Sex" value=""> Not specified
```

Note the use of value="" for the third option.

Note: You might think that you can achieve the same effect without this third option, simply by leaving all of the other options unselected. However, that doesn't work, because if the user selects one of those options and then changes their mind, there's no way for them to de-select it.

Checkboxes

The final field type we'll describe is the checkbox, where the user just ticks a box, like this:



Yes, I have read the Terms and Conditions.

The code looks like this:

```
<input type=checkbox name="Terms" value="Yes"> Yes, I have read the Terms and Conditions.
```

The field name is "Terms" in this example (because it says name="Terms").

Changing the return value

In this example, the value "Yes" is passed back to you if they tick this box (because it says value="Yes"). You can change this to whatever you like.

If you omit the value="xxx" altogether, the word "on" is passed back to you.

Checked by default

If you would like the box ticked by default, like this:

Yes, I have read the Terms and Conditions.

add the word "checked", like this:

```
<input type=checkbox name="Terms" value="Yes" checked> Yes, I have read the Terms and Conditions.
```

Multiple Choices

It's also common to see a list of checkboxes grouped together, like this:



What languages do you speak?

- French
- English
- Chinese

The main thing to understand about this is that the system doesn't have any understanding of such a grouping, and thinks of these as individual checkboxes. So it's up to you to group them in a logical way.

For example, the code might be:

```
What languages do you speak?
```

```
<input type=checkbox name="Language - French" value="Yes"> French  
<br><input type=checkbox name="Language - English" value="Yes"> English  
<br><input type=checkbox name="Language - Chinese" value="Yes"> Chinese
```

We're using three related field names ("Language – French", "Language – English" and "Language – Chinese"). You could equally use just "French", "English" and "Chinese"; but it would then be more difficult to recognise that they are part of the same group.

Layout guidelines

The description so far should be sufficient to help you create almost everything you require on a form. For the most part, you can add fields by simply copying existing fields (for example, from the code samples above) and modifying them.

This section describes some of the layout guidelines in more detail, if you are interested.



Line up fields

The fields are laid out in a two-column table so that they line up.

It usually has one field per row, with the field name in the left column and the space for them to type in the right column.

Example (which we've seen already):

First Name *

The code for that uses the input field type **text**, like this:

```
<tr><th align=left>First Name <font color="#ff0000">*</font></th>  
  <td align=left><input name="First Name" type=text size=30 required>  
</td></tr>
```

Stretch wide fields over two columns

However, if the field is very wide, putting it in the right column can distort the entire form, because there's not enough space for it. In this case, it's better to display the field below its field name.

An example is the Comments field shown above:

Comments:

To stretch it over two columns:



```
<tr><th align=left colspan=2>Comments:</th></tr>
<tr><td colspan=2><textarea name="Comments"></textarea></td></tr>
```

This isn't only for text blocks. It can also apply to radio buttons and drop-down menus if the option labels are very wide.

Use appropriate field sizes

Use the field size as a clue to the field contents. For example, set a smaller size for postcodes, like this:

```
<input type=text name="Postcode" size=8>
```

Identify required fields

Most forms have at least one compulsory field. Tell the user beforehand which fields are compulsory and which are optional. There are a number of ways of doing this:

- Put the word "(required)" next to compulsory fields (Use this if only a few fields are compulsory).
- Put the word "(optional)" next to optional fields (Use this if most fields are compulsory).
- Instead of using "(required)", it's also common to use an asterisk or a red asterisk to identify required fields (as in the sample form above). If you do this, explain what it means at the top of the form.



Error Page

If there are some required fields on the form (which is usually the case) and the user leaves them blank, the system displays an error page – e.g.

Incomplete Information

Sorry - we can't process your information:

- ◆ First Name: Please type a value
- ◆ Last Name: Please type a value

Please use the Back button on your browser now to return to the previous screen and fill in the details.

Thank you.

If you would like to change the wording on this page, create your own "blank_fields.html" using the E-Service Pages module. Copy this text for the Smart Text for the page:

```
Sorry - we can't process your information because you have left these fields blank:
```

```
--BLANKLIST--
```

```
Please use the Back button on your browser now to return to the previous screen and fill in the details.
```

```
Thank you.
```

Then adapt it to match the wording you would like.



Follow-Up Page

After the user submits the form, it takes them to a “follow up” page - usually something that thanks them or prompts them for further action.

You have three options for specifying the follow-up page:

- A page on your Web site.
- A page on your E-Service account.
- A default page.

Page on your site

To take them to a page on your Web site, use the hidden field “followup_url”, specifying the page name.

For example, if you create a page called “enquiry-thanks.html” on your site, set it like this:

```
<input type=hidden name="followup_url" value="enquiry-thanks.html">
```

Note that this must be a page on **your** Web site. The system automatically puts your Web address in front of the page name you specify here.

Page on E-Service site

If you are using the E-Service “pages” module, you can direct them to a page in your E-Service account by specifying it in the hidden field “followup_screen”, like this:

```
<input type=hidden name="followup_screen" value="enquiry-thanks.html">
```



The reason for using this option, rather than the first option, is because you can customise the output further:

- E-Service automatically replaces each occurrence of "--Blah--" with the value of the field named "Blah" from the form. For example, if you have a field called "First Name", you can say "Thank you, --First Name--, for your response".
- It substitutes "--date--" with the current date and time.

Default page

If you don't use either of the options above, it simply displays a basic "Thank You" page.

In the latter case (i.e. taking them to a page in your E-Service account), you can customise the output page further.

Testing Your Form

You can test the form as often as required. Just make sure that the hidden field "mail_to" is set to your own e-mail address. If this isn't the case, change it temporarily and remember to change it back later.

Here are some of the things you should consider testing:

These are the things to test:

- Fill in **everything** in the form and submit it. Ensure that the follow-up e-mail comes through correctly and includes **all** the fields.
- Ensure that all the field names are meaningful.



- Make sure that it displays the correct follow-up page.
- If there are any required fields, leave everything blank and submit the form. It should prompt for the required fields.
- Ensure that the results are stored for downloading (see Downloading Results below).
- Test the form in a 800x600 window (and preferably even a 640x480 window) to ensure that it all fits.
- Also test it in a wider window to ensure that it doesn't stretch out too much.

When you first create the form, you will probably run a number of tests until you get it right. After that initial testing, you might choose to delete the saved results so that they don't get mixed up with the real form submissions. You can do this as described below in the section about downloading results.

Downloading Results

The form results are saved on the E-Service Web site for you to download later.

Click the "Download feedback" link on the E-Service administration page:



This will show you a list of possible results to download:



Count	Title	From	To	Delete?
21	Affiliate Referrals	08-Jan-2004	17-Jan-2004	<input type="button" value="Delete"/>
59	AntiSpam Quiz			<input type="button" value="Delete"/>
1	AntiSpam Quiz Popup	23-Sep-2004	23-Sep-2004	<input type="button" value="Delete"/>
20	camtech	21-May-2003	25-Jul-2003	<input type="button" value="Delete"/>
3892	CourseBot Enrolments	31-May-2003	22-Oct-2004	<input type="button" value="Delete"/>
5	EBook Outline	05-Apr-2004	20-Apr-2004	<input type="button" value="Delete"/>
31	EBook Survey	16-Aug-2004	26-Aug-2004	<input type="button" value="Delete"/>
3	Ecommerce Commissions			<input type="button" value="Delete"/>
7	esec	05-Oct-2004	07-Oct-2004	<input type="button" value="Delete"/>

The list might include downloadable results from other E-Service modules as well – such as on-line orders, survey results and e-commerce transactions. They are all handled in the same way.

The columns in this list are as follows:

- Count: The number of results (in other words, the number of times the form was submitted).
- Title: The title of the form. Click this to display the form results and download them. Instructions for doing this are at the bottom of the page, below the list.
- From/To: The date of the earliest and most recent form submissions.
- Delete?: Click this link to delete that sort of form results. **WARNING:** This deletes the results permanently, so make sure you have downloaded them first.

The list is sorted by the Title column, but you can click other column headings to change the sort order.



Other Options

On-line courses

If you're using the CourseBot module, you can also provide optional checkboxes to allow the user to subscribe to various courses or newsletters.

For example, you might create a general "Contact Us" form where you also give the user the option of subscribing to a newsletter:

Yes, I would like to join the newsletter mailing list

To make this happen, you require the following basic fields:

- The person's first name, in a field called "First Name" or "f_first_name".
- The person's last name, in a field called "Last Name", "Surname" or "f_last_name".
- The person's e-mail address, in a field called "E-Mail", "email", "f_email" or "E-Mail Address".
- If any of the courses require a password, you also require a field called "Password", "password" or "f_password".

Then you require a checkbox for each of the courses that are being offered. The field name for the checkbox is "course-BLAH", where "BLAH" is the course code. For example, a typical line might be:

```
<input type=checkbox name="course-newsletter"> Join our mailing list
```

This assumes that there is a course called "newsletter".



Taking Orders

If you would like to include a reference/receipt number with your response, add a hidden field "reference", with a value containing the text "###", and the "###" is replaced by a generated number.

For example, if you have bookings for monthly meetings, you might have something like this:

```
<input type=hidden name="reference" value="Monthly###">
```

and the generated reference number might be, say, "Monthly03030695" (Technically, the reference number is the date in yymmdd format followed by two random digits).

The generated reference number can be displayed on the follow-up page (as described above). It's also passed to you (or whoever) in the follow-up e-mail.

E-mailing the User

Usually, only you get a copy of the form input submitted by the user. However, there are times when it's useful for the user themselves to have a copy of what they submitted. You can do this by adding this line:

```
<input type=hidden name=metoo value=1>
```

Using Other Web Design Software

This entire guide assumes that you're working in Smart Text on a Fast Forward Web site developed by First Step Communications. If you're not, you can still use the form and survey software if you have paid for it; but the process of creating the form has some small differences.



This section describes the minor differences that you have to manage if you're creating the form using your own Web design software.

List of fields

If you're not using Smart Text, add this field to your form:

```
<input type=hidden name=allfields value=1>
```

This tells the module to get all the field names starting with a capital letter, and use that as the list of fields.

Required fields

If there are any fields that are compulsory, they must appear in the hidden field "required_fields", which is a comma-separated list of field names.

The way this was described above was to insert the word "required" in the field, like this:

```
<input type=text name="Blah" required>
```

This will automatically create the "required_fields" field.

However, this won't work if you're using your own Web design software! In this case, you have to do it by adding a separate field like this:

```
<input type=hidden name="required_fields" value="Name,E-Mail Address">
```





More Help

- Frequently Asked Questions: <http://first-step-support.info/>
- Help Desk <http://firststep.com.au/help-desk.html>
- Other support options: <http://firststep.com.au/support.html>
- Consulting: <http://gihanperera.com/consulting.html>
- Products: <http://gihanperera.com/products.html>
- Other User Guides: <http://www.firststep.com.au/fast-forward-guides>



Subscribe

Subscribe to the e-mail newsletter *Expert's Gold* at www.firststep.com.au.

© Copyright 2009 First Step Communications

About First Step

Briefly, we build Web sites for speakers.

More fully, at First Step Communications www.firststep.com.au, we help information experts – speakers, trainers, consultants, coaches, authors, service professionals and others like them – to leverage their expertise on the Internet.

Disclaimer

The material contained in this report is general and is not intended as advice on any particular matter. The author expressly disclaims all and any liability to any persons whatsoever in respect of anything done by any such person in reliance, whether in whole or in part, on this report. Please take appropriate legal advice before acting on any information in this report.

