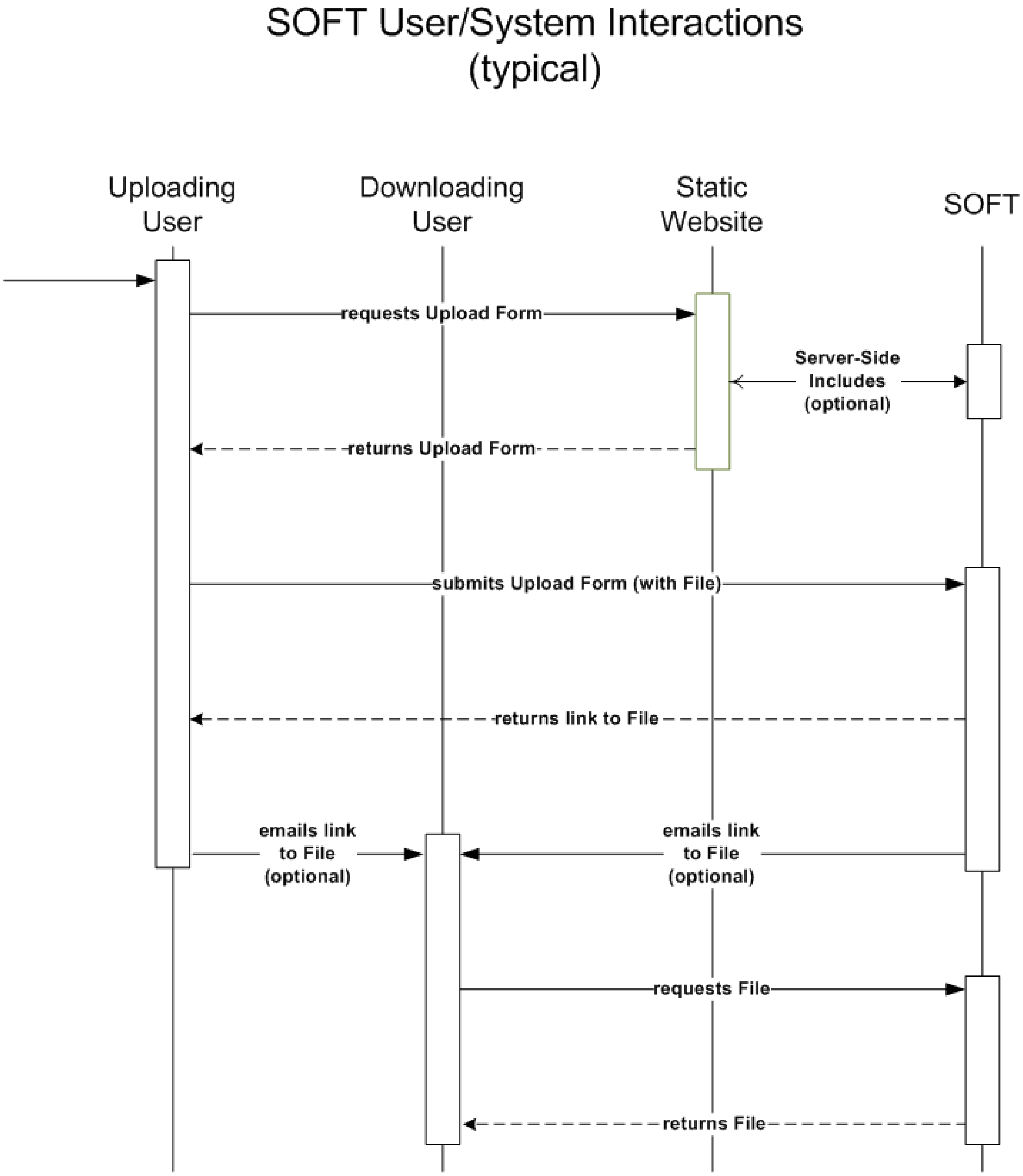
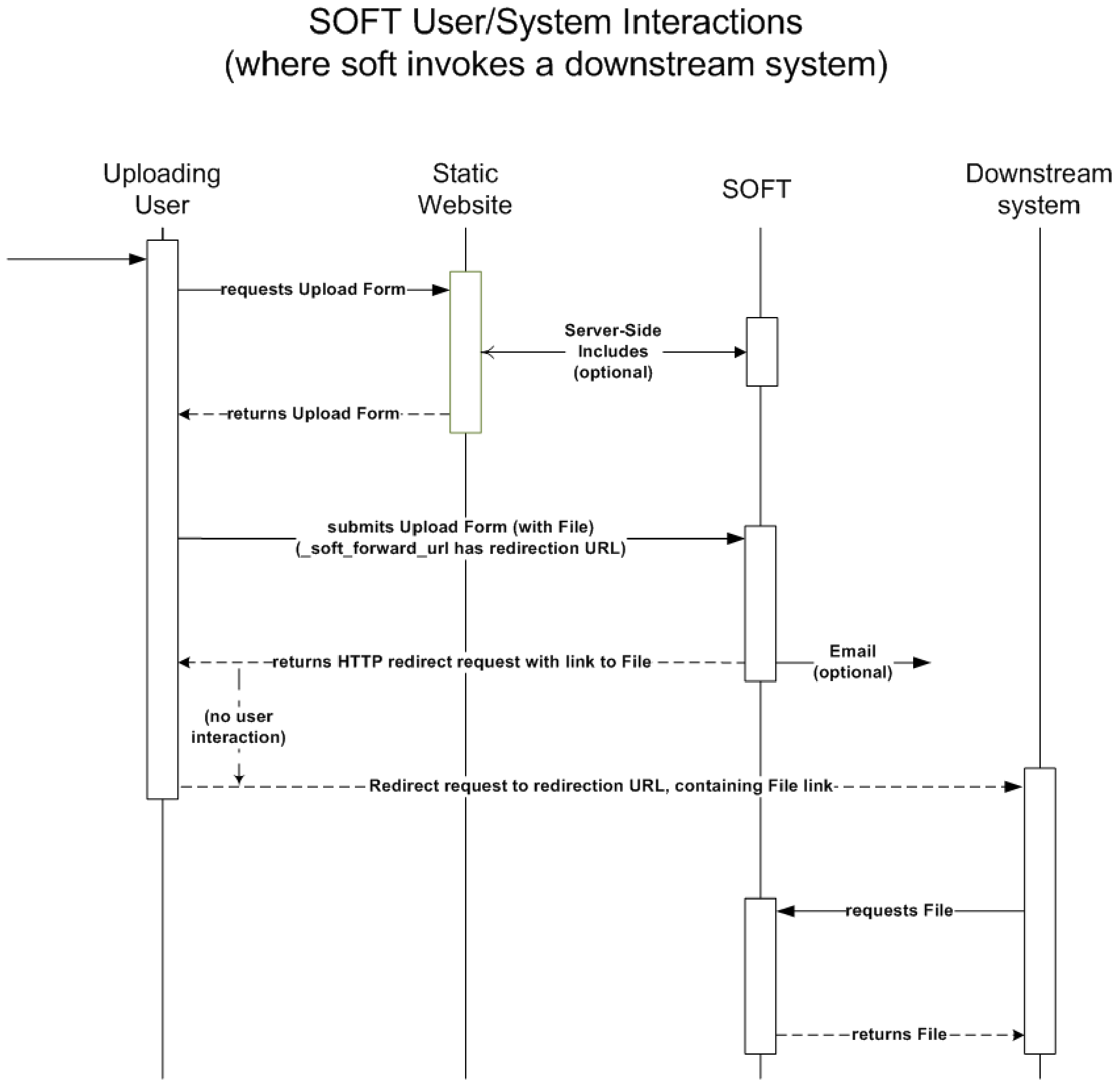
Simple Obfuscated File Transfer [SOFT] Developers Guide

SOFT version 1.10

SOFT is a utility that provides file transfer services. This guide provides information for developers who wish to build web page forms that use this service.

Upload requests come via an HTTP POST request containing specific parameters. A download URL is returned to the user, which is then used to download what was uploaded.


SOFT User/System Interactions
(typical)
Uploading User
Downloading User
Static Website
SOFT
requests Upload Form
Server-Side Includes (optional)
returns Upload Form
submits Upload Form (with File)
returns link to File
emails link to File (optional)
emails link to File (optional)
requests File
returns File


SOFT User/System Interactions
(where soft invokes a downstream system)
Uploading User
Static Website
SOFT
Downstream system
requests Upload Form
Server-Side Includes (optional)
returns Upload Form
submits Upload Form (with File)
(_soft_forward_url has redirection URL)
returns HTTP redirect request with link to File
Email (optional)
(no user interaction)
Redirect request to redirection URL, containing File link
requests File
returns File

Calling SOFT

SOFT must be called by an HTML web form with specific parameters (see below). The web form can be implemented on any web server, although certain helper-tools (server side includes) available for use in the forms will only work if the form is deployed on the same server as SOFT (i.e. the web servers used by Ministries of Environment, Agriculture and Lands, and the Environmental Assessment Office).

The simplest code snip that would call SOFT looks like this:

```
<form action="/perl/soft/ul.pl"
  method="post" enctype="multipart/form-data" >
  <p>File to Upload:
    <input type="file" name="_soft_filename" />
  </p>
    <input type="submit" class="button"
      name="_soft_submit" value="Upload File" />
</form>
```

The `form action` points to SOFT. In the example above, the form is on the same server as SOFT. If the form is on a different server, then the `form action` would need to point to SOFT on the SOFT server (for example, `form action="http://www.env.gov.bc.ca/perl/soft/ul.pl"`).

Additional parameters can be used to control the behavior of SOFT or to collect business-specific information.

SOFT Parameters

SOFT looks for the POST parameters listed below. Any further parameters that begin with _soft_ are ignored. Any parameters that do not start with _soft_ are assumed to contain business information specific to the upload form, and are bundled into README.TXT and metadata.xml files that accompany the file for download.

- `_soft_filename` (mandatory): the file on the browser's client to upload. If the file exceeds configured maximum filesize then an error is returned. If the file is unreadable or does not exist then an error is returned.
- `_soft_mail_sendto` (optional): an email address. If included and is valid then the system will send the recipient an email notification containing the URL of the uploaded file.
- `_soft_mail_subject` (optional): a subject line to use when sending the email. Default is something like: "A file is ready for download". These metasymbols can be used in the subject:
    - [f] is replaced with the name of the uploaded file;
    - [s] is replaced with the size of the uploaded file (although the file, once zipped, may be significantly smaller on the server and when downloaded)

  ```
  <input type="hidden" name=_soft_mail_subject
  value="File [f] is ready, size is [s]" />
  ```
- `_soft_keep_days` (optional): the number of days to retain the file before deleting from the transfer server. If the number of days exceeds the configured maximum then the maximum is used.
- `_soft_force_gov` (optional): Uploads from inside BC Government are by default available for download anywhere on the Internet. This parameter forces BC Gov uploads to be accessible only from within BC Gov, if it is set to any non-blank, non-zero value. This has no effect on uploads from outside BC Gov.
- `_soft_submit` (mandatory): this is the name of the submit button.
- `_soft_form_owner` (optional but recommended): an email address of the person who maintains the web page containing the form. This allows notifications related to SOFT maintenance or changes to go to business areas that rely on the service, and is only used by systems staff. E.g.

  ```
  <!-- Contact info for this form for SOFT maintenance notifications -->
  <input type="hidden" name=_soft_form_owner value="my.name@gov.bc.ca" />
  ```
- `_soft_forward_url` (optional): if set, on success SOFT will forward (redirect) the browser to the given URL. The file download link will be added to the URL as a parameter soft_url. This is intended to allow SOFT to pass its success page to a custom page or to pass to an information system for further processing. The 'forward URL' must be either HTTP or HTTPS, and reside on a .gov.bc.ca server. The URL may contain parameters, in which case the soft_url parameter is appended to the existing parameters.

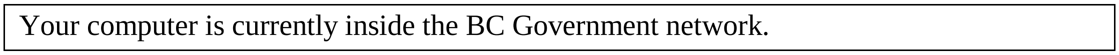
  ```
  <input type="hidden" name=_soft_forward_url
  value=”http://a100.gov.bc.ca/int/myapp/catcher.do?datatype=treefrogs”>
  ```

- `_soft_simple` (optional): if set true (Y/y/1) then sets a parameter in the retrieval URL that makes SOFT return only the original file. If set false (N/n/0) then sets a parameter in the retrieval URL that makes SOFT return the complete archive including README.TXT and metadata.xml (if available).
    - Note: a SOFT retrieval URL can force 'simple' download behavior by adding a parameter 'simple=y'; or force full retrieval by adding a parameter 'full=y' (or simple=n). E.g.:
      `http://big_long_soft_url?simple=y` will force download of only the uploaded file
      `http://big_long_soft_url?simple=n` will force download of the full archive.
      There may be a performance advantage to 'full' where SOFT has compressed a file, as in 'simple' mode downloads are not performed compressed.
- [other post parameters]: any other POST parameters are recorded in the archive's README.TXT file. Truncation and scrubbing may take place to assure that content is not tainted or does not exceed reasonable content limits. Note that there are limits to the amount of information that may be included in POST parameters though they're typically pretty generous. List items (e.g. from multi-picks) may or may not come through correctly [This is something to fix in a future release if required].

Server Side Includes

SOFT provides server-side includes that can be used in the creation of html forms.

- `inside.pl` – this returns a string saying if the user's computer is inside or outside the BC Government network. E.g.

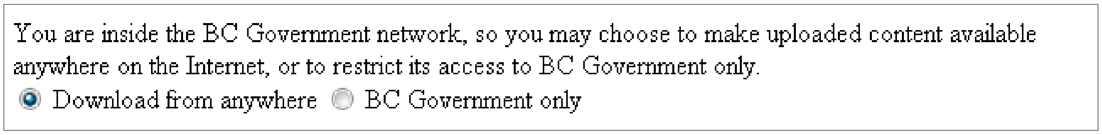
  

  Your computer is currently inside the BC Government network.

  It is called using a server-side include like this:

  ```
  <!--#include virtual="/perl/soft/inside.pl" -->
  ```

- `force_gov.pl` – if a user is outside the BC Gov, this does nothing. This returns a form element (radio button) to allow BC Government users to restrict access so that the uploaded file may be downloaded only by BC Government (if, for example, the file were being sent between employees). E.g.

  You are inside the BC Government network, so you may choose to make uploaded content available anywhere on the Internet, or to restrict its access to BC Government only.
  ● Download from anywhere ○ BC Government only

  It is called using a server-side include like this:

  ```
  <!--#include virtual="/perl/soft/force_gov.pl" -->
  ```

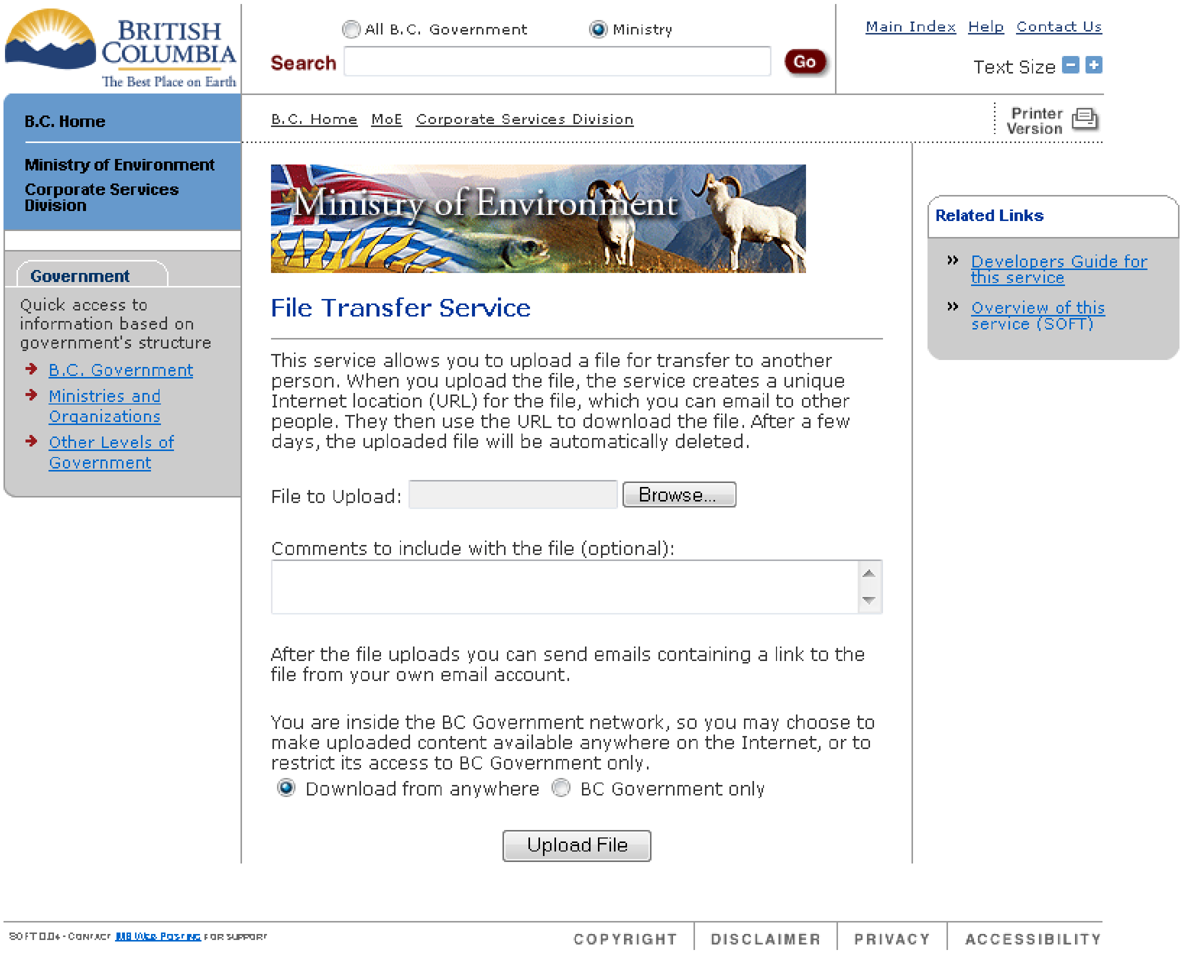
In order for a server side include to work, these conditions must be met:

1) The web form must be on the same web server as SOFT, and the call to the include must not include a server name (see example above)
2) The web form's filename must end in `.shtml` instead of `.html`
3) The directory containing the web form must have apache Includes enabled. This is done in the apache configuration file by adding `Options +Includes` for the area of the web server that contains the web form - contact IMB Web Hosting Services for support.

“Uploading” animation/message

It may be desirable to provide an animation or an “Uploading…” message while the file uploads. This is the responsibility of the web page that contains the web form. It is done using JavaScript, and in order to support the widest variety of browsers, this is the suggested implementation.

When the user pressed the ‘Upload’ button, part of the screen (including the ‘Upload’ button) should disappear, and be replaced by text and possibly an animated image (GIF). For example:



Quick access to information based on government's structure

File Transfer Service

This service allows you to upload a file for transfer to another person. When you upload the file, the service creates a unique Internet location (URL) for the file, which you can email to other people. They then use the URL to download the file. After a few days, the uploaded file will be automatically deleted.

File to Upload: Browse...

Comments to include with the file (optional):

After the file uploads you can send emails containing a link to the file from your own email account.

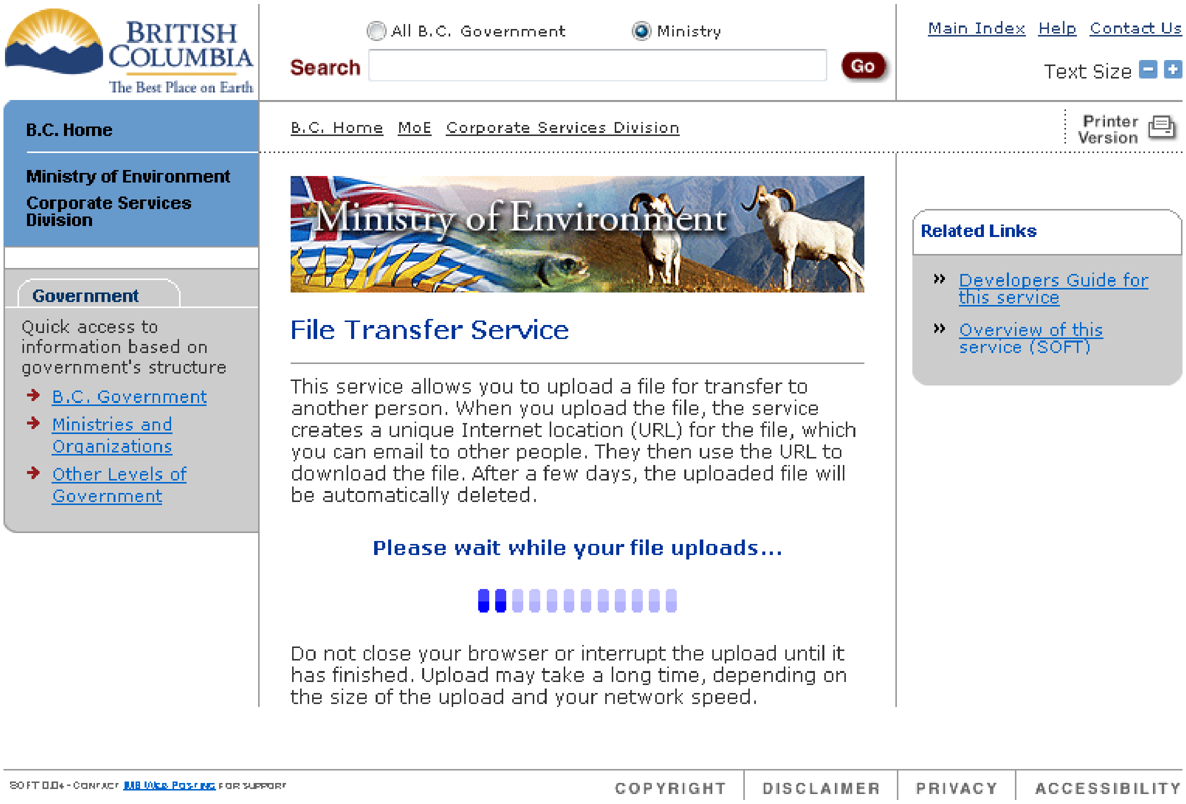
You are inside the BC Government network, so you may choose to make uploaded content available anywhere on the Internet, or to restrict its access to BC Government only.

● Download from anywhere ○ BC Government only

Upload File

SOFT 0.0.4 - Contact MB Web Posting for support

Becomes during upload, this (the form has been replaced by a message and an animation):



Quick access to information based on government's structure

File Transfer Service

This service allows you to upload a file for transfer to another person. When you upload the file, the service creates a unique Internet location (URL) for the file, which you can email to other people. They then use the URL to download the file. After a few days, the uploaded file will be automatically deleted.

Please wait while your file uploads...

Do not close your browser or interrupt the upload until it has finished. Upload may take a long time, depending on the size of the upload and your network speed.

In the HTML web form, this requires

- A short JavaScript function.
- Use of `<div>` tags to surround the areas to be shown while the form is completed (“submission-area” in the example below) & shown while the upload takes place (“upload-msg-area” in the example below).
- Modifications to the `<form>` and one of the `<input>` tags in the form.
- A <noscript> section that will only be displayed if JavaScript is disabled in the browser.

Example:

The Javascript portion should go before the first <div> begins.

```
<script type="text/javascript">
  // IE stops animating GIFs once a form Submit takes place, so
  // animated GIFs should be activated *after* the submit.  Arrange this
  // by calling this function by means of a setTimeout() call.  If the
  // setTimeout() is too short, then over slow connections it will still
  // be too soon.
  var showUploading = function() {
    document.getElementById('submission-area').style.display = 'none';
    document.getElementById('upload-msg-area').style.display = '';
  };
</script>
```

The first `<div>` identifies the area that will be visible while the user completes the form, and which will disappear when they press the ‘Upload’ button.

In this example, the entire form is inside this first <div>. Notice that inside the form,

1. An "`onSubmit`" has been added to the opening `<form>` tag.
2. For the <input> element that provides the 'Upload' button, on `onClick` has been added:
   `onclick="setTimeout(showUploading, 500);"`

```
<div id="submission-area">

  <form action="/perl/soft/ul.pl"
    method="post" enctype="multipart/form-data"
    onSubmit=
      "document.getElementById('submission-area').style.display = 'none';"  >

    <p>File to Upload: <input type="file" name="_soft_filename" /> </p>

    <p>Comments to include with the file (optional):<br>
    <textarea wrap="physical" cols="50" rows="2" name="comments"></textarea>
    </p>

    <p>After the file uploads you can send emails containing a link to the
      file from your own email account. </p>

    <!-- This will display radio buttons for _soft_force_gov parameter
         to users who are inside the BC Gov network -->
    <!--#include virtual="/perl/soft/force_gov.pl" -->

    <center>
      <input type="submit" class="button"
        name="_soft_submit" value="Upload File"
        onclick="setTimeout(showUploading, 500);" />
    </center>
  </form>
</div>
```

In the "upload-msg-area" `<div>`, put contents that will be displayed only after the user presses the 'Upload' button. In the example below, "wait4.gif" is the animated GIF image.

```
<div id="upload-msg-area" style="display: none;">
  <center>
    <h3>Please wait while your file uploads...</h3>
    <p><img src="wait4.gif" alt="" /></p>
  </center>
  <p>Do not close your browser or interrupt the upload until it has finished.
  Upload may take a long time, depending on the size of the upload and your
  network speed.  </p>
</div>
```

The `<noscript>` section will only be displayed if the user's browser has JavaScript disabled. It should warn them that not much will *appear* to happen after they click 'Upload'.

```
<noscript>
  <h3>Your browser has Javascript disabled.  </h3>
  <p>When you press the 'Upload File' button, there will be no
```

```
    indication that the upload is taking place; please wait and do
    not press the button a second time. </p>
</noscript>
```