

Tri-Tel HTTP/POST Fax Gateway
Tri-Tel Telecommunications
July, 2005

Tri-Tel Telecommunications offers several delivery methods for its faxing service. This document will describe the HTTP/POST delivery method. The HTTP/POST delivery method is a simple fax delivery method supporting only one recipient per fax, and only one document per fax. This API is provided to clients who wish to programmatically send faxes using a method other than email (SMTP).

The HTTP/POST delivery method is provided both under standard HTTP, and secured HTTP over SSL (HTTPS).

HTTP Request:

The HTTP request must be a POST request using the content-type: "multipart/form-data" in order to function properly. In HTML, this is done by including the parameters *METHOD=POST* and *ENCTYPE="multipart/form-data"* in the *<FORM>* tag. The form must call Tri-Tel Telecommunication's fax gateway script, located at:

<http://www.tri-tel.com.au/faxgateway.pl> (unsecure), or
<https://www.tri-tel.com.au/faxgateway.pl> (secure)

Fields:

The following table lists the fields available in the POST method. The fields marked 'required' must be present in order for the fax to be delivered.

Required	Name	Description
Yes	"mode"	A string specifying the type of operation. Options are: "verify" Verifies the content of the request, but does not send the fax. "send" Verifies and sends the fax request.
No	"subject"	A subject line for the fax job. This subject line will not appear on any faxed document. It only appears in the emailed fax reports.
Yes	"user"	The Tri-Tel account username
Yes	"pass"	The Tri-Tel account password. This field is case sensitive.
Yes	"recipient"	A string containing the destination fax number, in international format, without any spaces or symbols. A correct value would be "612901234567".
Yes	"file"	A file object. This must be an attached file.

File types:

The Tri-Tel Telecommunications fax service supports several file types. The PDF format is the most recommended as it preserves formatting, and embeds related images and fonts. However, Tri-Tel also supports Microsoft Office documents (Word, Excel, PowerPoint...), HTML, plain text and RTF documents. Support for other document formats may be added in the future. If a fax is sent in a format that is not supported, an e-mail will be sent in reply stating that the format is not supported.

Output:

After submitting a fax request, the result will be output to the client using the following format:

[HTTP Headers]

["Error" or "Success"] : [Error Code] : [Message]

Error codes:

- 200: No error – Fax accepted for delivery.
- 202: There is no recipient.
- 204: Your balance is too low. (Your current balance will appear in the message)
- 205: Your username or password is invalid; or the service is temporarily unavailable.
- 207: There was no attached file.
- 209: Request not understood; one or more of the required values are invalid or not present.
- 210: This service is currently disabled in your account. Please contact customer-service.
- 211: You are not currently subscribed to the faxing service.

If you are unsure of your form data, submit it using "mode"="verify" and check the error code. Correct your data until the error code is "200" (Ok), then submit your form again using "mode"="send".

Examples:

Success: 200: Ok

Error: 205: Not Authenticated

Error: 210: Service Disabled

Example form:

The following example is a simple HTML form that will allow a user to submit a fax using a normal Internet browser.

```
<FORM METHOD="POST" ACTION="http://www.tri-tel.com.au/faxgateway.pl"
  ENCTYPE="multipart/form-data">
<INPUT TYPE=HIDDEN NAME="mode" VALUE="send">

<TABLE BORDER=0>
<TR>  <TD>Username:                                </TD>
<TD><INPUT TYPE=TEXT NAME="user" VALUE="">        </TD></TR>
<TR>  <TD>Password:                                  </TD>
<TD><INPUT TYPE=PASSWORD NAME="pass" VALUE="">    </TD></TR>
<TR>  <TD>Subject:                                    </TD>
<TD><INPUT TYPE=TEXT NAME="subject" VALUE="" SIZE=40> </TD></TR>
<TR>  <TD>To Fax Number:                             </TD>
<TD><INPUT TYPE=TEXT NAME="recipient" VALUE="">   </TD></TR>
<TR>  <TD>File:                                       </TD>
<TD><INPUT TYPE=FILE NAME="file" VALUE="">        </TD></TR>
<TR>  <TD>                                           </TD>
<TD><INPUT TYPE=SUBMIT VALUE="Send">              </TD></TR>
</TABLE>
</FORM>
```

Username:	<input type="text" value="my_user"/>
Password:	<input type="password" value="*****"/>
Subject:	<input type="text" value="Your Invoice"/>
To Fax Number:	<input type="text" value="6172345678"/>
File:	<input type="text" value="Documents\invoice.pdf"/> <input type="button" value="Browse..."/>
	<input type="button" value="Send"/>

Perl Example:

The following perl script automatically sends a fax; then verifies and outputs the result.

```
#####
#!perl
# Tri-Tel Telecommunications

use HTTP::Request::Common;
use LWP::UserAgent;
use HTTP::Response;

print "Content-Type: text/html\n\n";

# Send the fax

my $errorcode =
    TTSendFax("my_username","my_password","subject","61291234567","C:\\test.pdf");

# Verify result

print "Success!\n"           if ($errorcode == 200);
print "Failed: No recipient.\n"   if ($errorcode == 202);
print "Failed: Balance too low.\n" if ($errorcode == 204);
print "Failed: Authentication error.\n" if ($errorcode == 205);
print "Failed: No attachment.\n"   if ($errorcode == 207);
print "Failed: General error.\n"   if ($errorcode == 209);
print "Failed: Service is disabled.\n" if ($errorcode == 210);
print "Failed: Not subscribed.\n"   if ($errorcode == 211);

exit();

#####
# Function TTSendFax:
# Parameters:
#   username,password,subject,number,file
#####

sub TTSendFax()
{
    # Get parameters
    my $username = shift;
    my $password = shift;
    my $subject = shift;
    my $recipient = shift;
    my $file = shift;

    my $status; my $code; my $message; my $head; my $body;

    # Send POST request

    my $browser = new LWP::UserAgent;
    my $result = $browser->request(
        POST 'http://www.tri-tel.com.au/faxgateway.pl',
        Content_Type => 'multipart/form-data',
        Content => [
            'user'           => "$username",
            'pass'           => "$password",
            'subject'        => "$subject",
            'recipient'      => "$recipient",
            'mode'           => 'send',
            'file'           => [$file],
        ]
    );

    # Extract error code
    ($head, $body)=split('\n\n', $result->as_string());
    ($status, $code, $message)=split(':', $body);

    return $code;
}
}
```