Hosted Web Form with Post Back Data

Version 2.2.2

Revision Date: November 10, 2014

Pre-Loading the Hosted Web Form Values

You can pre-load the following input fields in the online payment portal by supplying their values in the url:

1. Account	Your reference account information for this customer.
2. FirstName	Customer's first name.
3. LastName	Customer's last name.
4. Country	Values: USA or CAN
5. Address1	Line 1 of customer's address.
6. Address2	Line 2 of customer's address.
7. City	City where customer resides.
8. State	Two-character state abbreviation.
9. Province	Two-character province abbreviation.
10. ZipCode	Zip code (USA) or Postal code (Canada)
11. Email	Customer's email address
12. Phone	Customer's phone number
13. Amount	The payment amount. Do not include dollar-sign.
14. Comment	Any comment for the 'Comments' text-area
15. PaymentMethod	Values: CC = Card payment or ACH = ACH/E-Check /Bank payment
16. CardName	The name on customer's credit card
17. CardNum	The credit card number
18. ExpDate	Expiration date of card in format: MMYY
19. Cvv	The 3 or 4-digit card verification number
20. RoutingNum	Bank routing/ABA number
21. BankAcctNum	The customer's bank account number
22. BankAcctType	Values: C = Checking account or S = Savings account
23. [Custom Field]	Any custom field name

Notes:

The parameter names are not case sensitive: "Account", "account", "ACCOUNT" are all the same.

You can make any field in the Hosted Web Form a "read-only" field by providing a value for it with a tilde (~) symbol as the first character. The form will remove the leading tilde, load the value into that field, and make that field read-only.

If a postback url has been specified, all of the parameters in this request will be passed back in the postback data in the "arglist" parameter (see below.)

Pre-Loading Custom Fields

Any merchant-defined custom fields can be pre-loaded with information. The read-only behavior using (~) also applies to custom fields.

Custom fields are named in the following way:

“Dept” + <departmentid> + “_” + <custom field name>

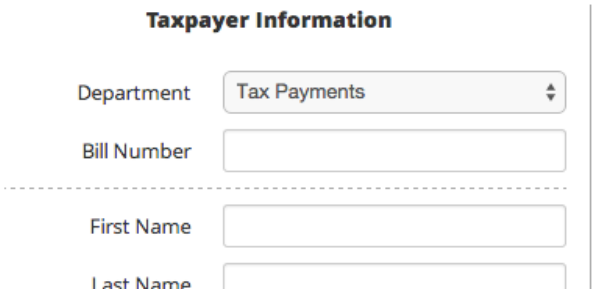
For example a custom field named “BillingCode" in department 88 would appear as: “Dept88_BillingCode”

In order to preload this field the url would look something like this:

https://secure.cpteller.com/terminal/portal/?op=Key123&Dept88_BillingNumber=9817239

It may be necessary for a merchant to view the page source generated by their hosted web page in order to determine the actual (mangled) name of the data field they want to pre-load. The following steps explain how this can be done.

1. Open the hosted online payment page
2. Right-click on the page and select “View Page Source”
3. Search for the field label. For example: “Bill Number”
4. The name of the field is inside the “input” angle-bracket in the “name” parameter. (See image.)
5. In this example it can be determined that the name of the “Bill Number” item in the page is: “Dept88_BillNumber”
6. Test this by modifying the url in step 1 like so: https://secure.cpteller.com/terminal/portal/?op=Key123&Dept88_BillNumber=123
7. Verify that “123” appears in the “Bill Number” box

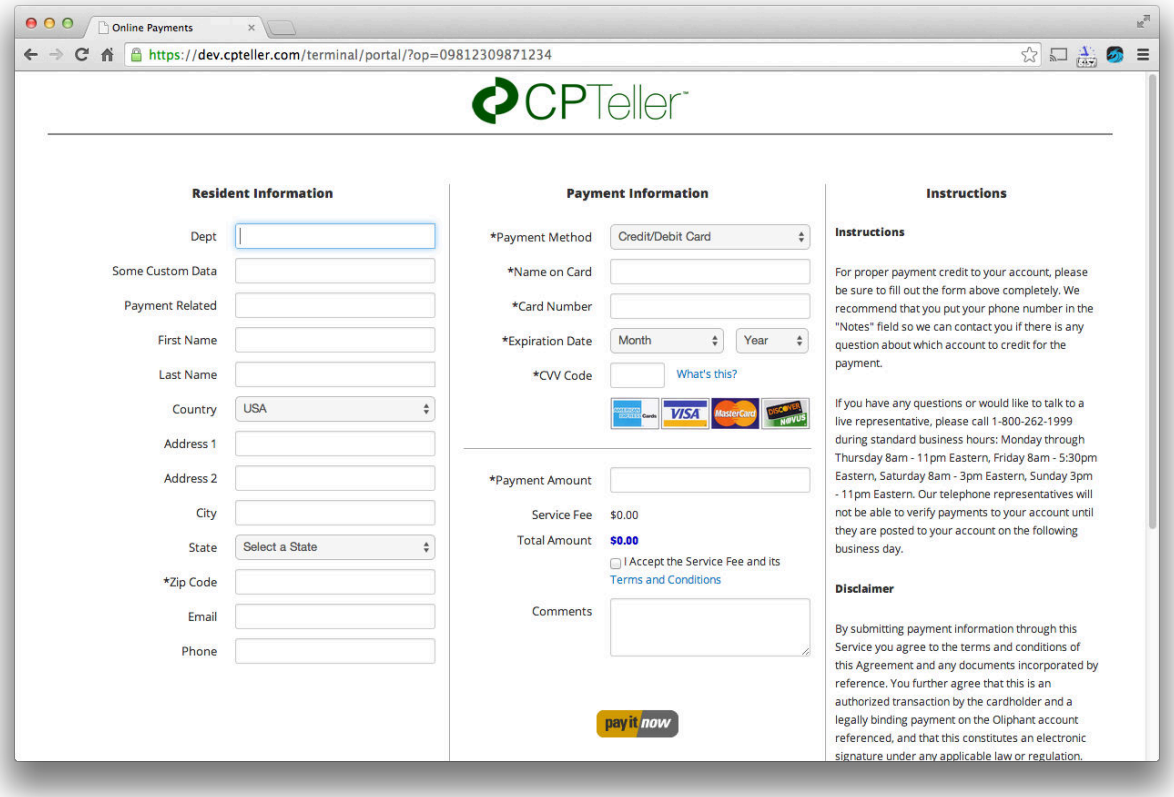


```
</div>
<div id="Dept88">
<div class="control-group">
<label class="control-label" for="Dept88_BillNumber">Bill Number</label>
<div class="controls">
<input type="text" class="input-large" name="Dept88_BillNumber" id="Dept88_BillNumber" value="" maxlength="100">
</div>
</div>
</div>
<div class="control-group>
<div class="controls" style="border-top: 1px dashed #AAA; margin-bottom: 10px; margin-left: 20px;">
```

Hosted Web Form Setup

There are a number of customizations you can perform on the online payment page:

1. The account field label is customizable to anything you want it to say: “Student ID”, “Social Security Number”, etc.
2. The banner image at the top of the page can be anything you like.
3. A return-to url link can be added to the page in case someone wants to go back to (wherever you specify) instead of making a payment.
4. The submit button text is customizable, and a custom submit button image may be uploaded and enabled on the page.
5. The Instructions can be anything you want it to be.
6. Underneath the first field (Account) and before the “First Name” field you can add any number of totally custom fields. These are for your own use.
7. A “Department” feature is available for you to define some shopping cart-like features so that your customers can choose from a dropdown list of pre-defined items you specify.
8. You may specify which credit card images to show to the user.



This screenshot shows an online web form with “Dept” as the Account field label, a custom data field with “Some Custom Data” label, a second custom data field with “Payment Related” as a label, a custom button to submit the payment, and “Service Fees” enabled.

Postback Data

If you specify a post back url the page will post all of the relevant data from the form to that url. The following items are posted back:

1. Op: This is your online payment page key. It is what is used to identify your online payment terminal so that it will show your company's banner image and so that payments made on the terminal will fund to your company.
2. Timestamp: The time the payment occurred (Mountain Time Zone.)
3. Customerid: The id or token, which has been assigned to your customer as a result of the payment.
4. Paymentid: The id or token, which has been assigned to the payment.
5. Account: The account information as it was entered on the page.
6. Firstname: Your customer's first name as they entered it on the page.
7. Lastname: Your customer's last name as they entered it on the page.
8. Address1: First line of customer's address as entered.
9. Address2: Second line of customer's address as entered.
10. City: Customer's city as entered.
11. State: Customer's state as entered.
12. Zipcode: Customer's zipcode as entered.
13. Phone: Customer's phone number as they entered it on the form.
14. Department: The department text the user selected if departments are enabled.
15. Method: "CARD" or "ACH"
16. Invoice: The invoice field is currently automatically assigned a value.
17. Authcode: The authorization code if a card payment.
18. Avsdata: The address verification system result for a card payment.
19. Ipaddress: The IP Address from which the user made the payment.
20. Amount: The amount of the payment.
21. Fee: The service or convenience fee if any.
22. Total: The amount of the payment plus the fee (if any.)
23. Comment: Any comment the user typed in the Comments field on the form.
24. <custom data field name>: The custom data fieldnames you define in setup, and their values as entered by the user will be included at the end of the data.
25. ArgList: A list of all name/value arguments received in the online payment page request.*

Notes:

For ACH payments authcode and avsdata will be empty.

* The arglist is returned as a list of name=value pairs separated by the verticals bar "|" character (ASCII 124.) This should make it easy for merchants to include any extra parameters in their requests (such as tracking numbers.) The arglist may be enclosed in double-quotes.

Example

Here is some sample code, which "catches" post back data. It is written in cfscript. (It is actual working code but you can think of it as pseudo-code.) The sample code does nothing other than write the information to a log file. (You will probably want to do more than this.)

```
<cfscript>

  var logfile = new lib.logfile("/log/opterm.log");
  logfile.writeline("This is the postback url.");

  logfile.writeline("op: " & op);
  logfile.writeline("timestamp: " & timestamp);
  logfile.writeline("customerid: " & customerid);
  logfile.writeline("paymentid: " & paymentid);
  logfile.writeline("account: " & account);
  logfile.writeline("firstname: " & firstname);
  logfile.writeline("lastname: " & lastname);
  logfile.writeline("address1: " & address1);
  logfile.writeline("address2: " & address2);
  logfile.writeline("city: " & city);
  logfile.writeline("state: " & state);
  logfile.writeline("zipcode: " & zipcode);
  logfile.writeline("email: " & email);
  logfile.writeline("phone: " & phone);
  logfile.writeline("department: " & department);
  logfile.writeline("method: " & method);
  logfile.writeline("invoice: " & invoice);
  logfile.writeline("authcode: " & authcode);
  logfile.writeline("avsdata: " & avsdata);
  logfile.writeline("ipaddress: " & ipaddress);
  logfile.writeline("amount: " & amount);
  logfile.writeline("fee: " & fee);
  logfile.writeline("comment: " & comment);
  logfile.writeline("arglist: " & arglist);

</cfscript>
```

…and here is the output from the sample code:

```
09/29/14 05:23:18->arglist: op=09812309871234
09/29/14 05:23:18->postbackurl: http://localhost:8080/test/postback.cfm
09/29/14 05:23:18->This is the postback url.
```

09/29/14 05:23:18->op: 09812309871234
09/29/14 05:23:18->timestamp: {ts '2014-09-29 17:23:18'}
09/29/14 05:23:18->customerid: 1547492
09/29/14 05:23:18->paymentid: 272951
09/29/14 05:23:18->account: test
09/29/14 05:23:18->firstname: John
09/29/14 05:23:18->lastname: Moss
09/29/14 05:23:18->address1:
09/29/14 05:23:18->address2:
09/29/14 05:23:18->city:
09/29/14 05:23:18->state:
09/29/14 05:23:18->zipcode: 84092
09/29/14 05:23:18->email:
09/29/14 05:23:18->phone:
09/29/14 05:23:18->department:
09/29/14 05:23:18->method: CARD
09/29/14 05:23:18->invoice: C1547492
09/29/14 05:23:18->authcode: 929980
09/29/14 05:23:18->avsdata: N
09/29/14 05:23:18->ipaddress: 0:0:0:0:0:0:0:1
09/29/14 05:23:18->amount: 1.00
09/29/14 05:23:18->fee: 2.00
09/29/14 05:23:18->comment:
09/29/14 05:23:18->arglist: "op=09812309871234|trackingnum=9817293"