**Accessing Checkout Example**

This is an example of a web page which accesses the Checkout module:

A customer accesses conference information from the Psychology Department website. The customer is

going to pay for a Psychology Department Conference ($124.00) and lunch at the conference ($15.60).

When the customer clicks the button labeled “Make Payment”, he or she will be redirected to the

CASHNet.com website to enter payment information and confirm the transaction.

<html>

<head>

<title>Our Storefront</title>

</head>

<body>

<form method="get"

action="http://commerce.cashnet.com/MyUnivCheckout">

<input type="hidden" name="itemcode1" value="PSYCH-CONF"><br>

<input type="hidden" name="amount1" value="124.00"><br>

<input type="hidden" name="itemcode2" value="PSYCH-LUNCH"><br>

<input type="hidden" name="amount2" value="15.60"><br>

<input type="submit" name="submit" value="Make Payment">

</form>

</body>

</html>

Many other fields can be included in the query to pass additional information about the customer and the items being purchased. Some of the fields may be hidden fields (or supplied programmatically), while others may be text fields being entered by the student/customer. It makes no difference where they come from or how you mix them up; that is up to the programmer of each website. Some of the fields are optional, such as address. However, if you have the data available, you may want to pass it as well so that the information can be pre‐filled in the checkout screen reducing the amount of data entry required by the customer.

For each item code you can either send the amount and the description from the web form or you can have the system send the amount and description stored in the CASHNet item code table. For security reasons, Higher One Payments recommends that the second method be used whenever possible. If there are too many items to set up in CASHNet or if the price is determined on the fly necessitating the use of the first method, make sure the website is structured so that the customer cannot access the form.

**Transferring users back to the application**

**Information**

This section is only applicable if you are planning to use CASHNet Checkout for the integration.

Once the transaction is completed in CASHNet, users can be transferred back to your application through a specified sign‐out URL.

The URL can be configured in CASHNet store setup as a static field or can be passed as a parameter

(signouturl) to CASHNet during the initial transfer to CASHNet. The information to be posted back to the specified URL is sent in standard query format and can be sent as HTTPS POST or GET.

The information passed back is similar to store notification, however the sign‐out URL provides flexibility on the number of the parameters that may be sent. Please refer to **Appendix B: Response and**

**Notification Parameters** for more details. CASHNet provides a separate sign‐out URL for unsuccessful transactions as well.

**CASHNet Reference Types**

CASHNet reference types are custom properties that you can assign to the whole transaction or to individual items in a transaction. There are two types of reference types: transaction‐level references and Item‐level references. Transaction‐level references are attached to the whole transaction while

Item‐level references are attached to a specific item in the transaction. These two types of references are passed to CASHNet Checkout in different ways (see below). Both types of references can echo back with Checkout notification.

**Passing references to CASHNet Checkout**

References are passed along with other Checkout parameters. Transaction‐level references are passed with reference type names as the standard query name‐value pair, while item‐level references are passed as two name‐value pairs in order to identify the item code to which these references are attached.

Following are some examples.

**Example 1: Sending two transaction level references**

This example sends two transaction level references (SESSIONID and USERID) to CASHNet Checkout.

https://commerce.cashnet.com/chkoutx?itemcode=FEE&amount=123.00&SESSIONID=2334ffa&USER

ID=123446

**Example 2: Sending one item level reference for an item**

This example sends an item code reference. The name and value are passed separately through ref*<ref\_index>*type*<item\_index>* and ref*<ref\_index>*val*<item\_index>*, where *item\_index* is the index of the item and *ref\_index* is the index of the reference type being passed.

https://commerce.cashnet.com/chkoutx?itemcode=FEE&amount=123&ref1type1=SESSIONID&

ref1val1=2334f

**Example 2: Sending two item level reference for an item**

This is an example of sending multiple item‐level references.

https://commerce.cashnet.com/chkoutx?itemcode=FEE&amount=123&ref1type1=SESSIONID&ref1val1=2334ffa&ref2type1=USERID& ref2val1=123446

**Information**

You can send the item code references in any order; it has no relationship on what order it is configured in

CASHNet. For example, you can send the same data in the above example as shown below with exact same result.

http://commerce.cashnet.com/chkoutx?itemcode=FEE&amount=123&ref1type1=USERID&

ref1val1=123446&ref2type1= SESSIONID& ref2val1=2334ffa

**Receiving Payment notification**

Once the payment transaction is completed, CASHNet can send the payment notification as described earlier in this document and these custom parameters can be echoed back to your notification end point.

If you are using extract, synchronized exact or sign‐out URL as the notification method, you have flexibility in determining the parameter names for the reference values being echoed back. But if you are using the store notification method for sending these parameters, the parameter syntax is fixed.

Please refer to **Appendix B: Response and Notification Parameters** for more details

**PrePopulating Values**

If you are using Checkout for payment integration, when the user is redirected to CASHNet to take payment he will be presented with a form to enter credit card information, including credit card number, name on the card, billing address, etc. In many scenarios, your application may have either maintained or already collected much of this information. CASHNet provides a way to pre‐populate the form fields by passing this information along with other Checkout parameters.

These values can be passed as an item‐level reference for the first item you are passing. Here is an example of pre‐populating the card name, address, city, state, zip and email address field through

Checkout parameters.

https://commerce.cashnet.com/chkoutx?itemcode=FEE&amount=123&ref1type1=CARD\_NAME\_G&ref

1val1=John Foo&ref2type1=ADDR\_G&ref2val1=My

address&ref3type1=CITY\_G&ref3val1=City&ref4type1=STATE\_G&ref4val1=CA&ref5type1=ZIP\_G&r

ef5val1=123445&ref6type1=EMAIL\_G&ref6val1=a@g.com

As shown in the example above, this information is passed through the following special item‐level references:

􀂃 CARD\_NAME\_G : Name on the card

􀂃 ADDR\_G: Address

􀂃 CITY\_G: City

􀂃 STATE\_G: State

􀂃 ZIP\_G: Zip code

􀂃 EMAIL\_G: Email address

**Checkout – Payment notification**

Once the payment process is completed as described in the previous sections, CASHNet can notify your application through various methods. If you have chosen to get notified using an online/batch extract or synchronized extract, almost all the information about the transaction may be returned to your application in the format you desire, including transaction number, type of card, authorization code, custom parameters, etc. If you have opted to receive the notification by store notification, however, the parameter set and name are fixed.

Listed below are the details of the payment notification if you are using the store notification method.

