Digital Buy Online Shopping Cart Integration Overview

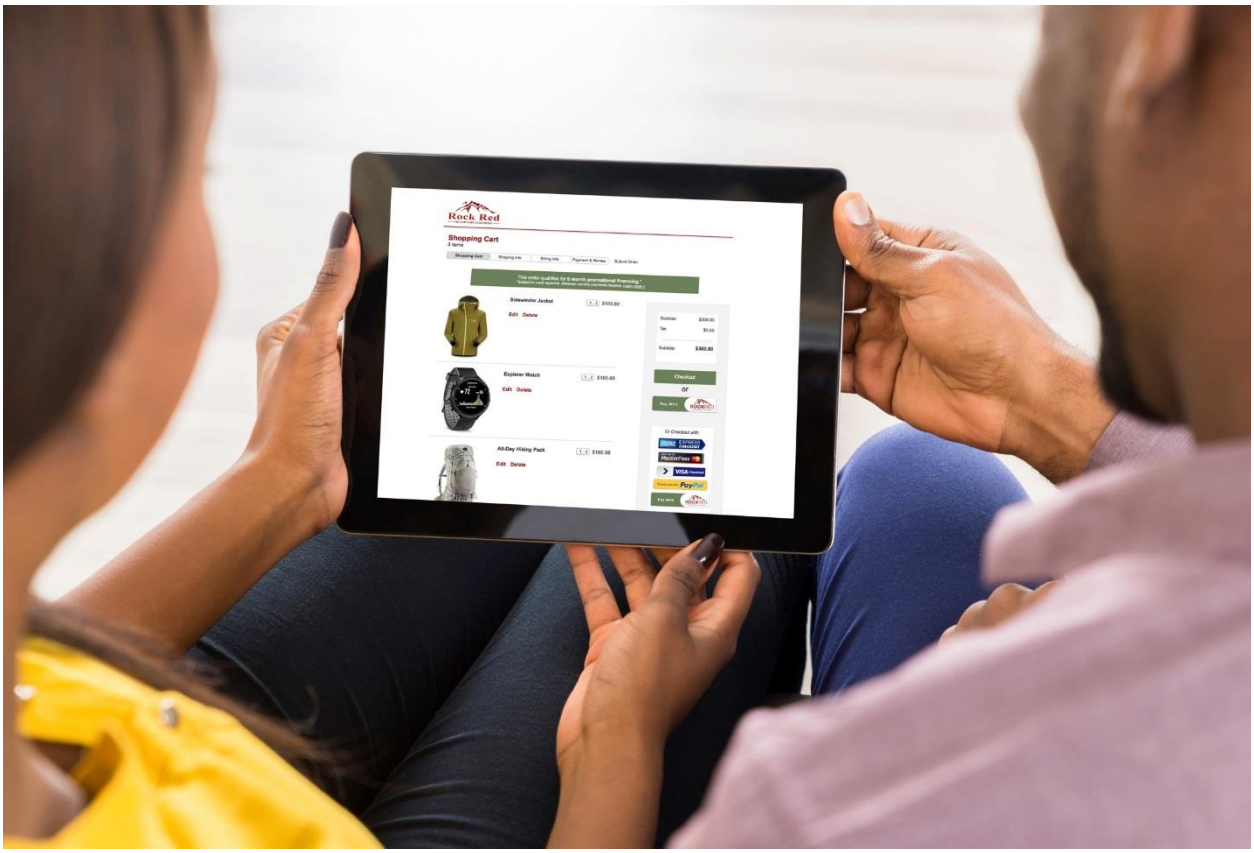
Version: **06-21-2018**


DIGITAL BUY
INTEGRATION OVERVIEW

TABLE OF CONTENTS

1 Introduction 5
2 Integration Highlights 6
3 Shopping Cart Setup 7
3.1 Cart Level Button 7
3.2 Special Financing 8
3.3 Settlement/Funding 10
4 Implementation 11
4.1 Integration Checklist 12
4.2 Website Setup 13
4.2.1 Recommended Checkout Flow 13
4.2.2 Digital Buy Source Files 17
4.3 Authentication 18
4.4 Consumer Account Information Modal 23
4.4.1 Consumer Account Information Modal Execution 23
4.4.2 Consumer Account Information Modal Response Handling 26
4.4.3 Returned Customer Address Information 32
4.5 Order Completion Modal 33
4.5.1 Order Completion Modal Execution 33
4.5.2 Order Completion Modal Response Handling 36
4.6 Combined Modal 41
4.6.1 Combined Modal Execution 42
4.6.1 Combined Modal Response Handling 45
4.7 Callback Status Messages 53

4.8 Status Inquiry API 54
4.8.1 findStatus API Response Parameters 55
4.9 Callback Redirection 58
5 Troubleshooting 59
5.1 Authentication Attempts Failing with '401-Unauthorized' 60
5.2 Not Receiving Responses ("Postbacks") 60
6 Glossary of Terms 61
7 License Terms 65



1 Introduction

Digital Buy is an online payment platform that makes accepting Synchrony Financial (SYF) issued credit cards easy and intuitive for customers. This guide will explain the features of Digital Buy and how to integrate it into an online shopping cart.

2 Integration Highlights

Digital Buy offers a lean, compliant, and secure method for accepting Synchrony Financial credit cards.

Integration with Digital Buy is accomplished with commonly used web technology, and can be logically broken into four parts.

- The first part of the integration is to include a set of JavaScript and CSS files on your site via the links provided in the Digital Buy implementation guide. These files are used to present the Digital Buy modals to the customer.

- The second part is the ability to provide the merchant username and password in a secure manner to the Digital Buy system, a process referred to as authentication. Authentication occurs for each checkout instance, and successful authentication results a unique pair of tokens for each specific customer checkout experience. This is done through a simple secure POST to Digital Buy, which will return the tokens in JSON (JavaScript Object Notation) format.

- The third part includes adding customized checkout buttons to your website which will initiate the SYF hosted customer-facing modals. These calls are made by submitting a form with the client to a JavaScript method with the client Token receive from the authentication. The last part necessary for implementing Digital Buy is to set the ability to parse a Callback Message (optional) and call a Status Inquiry API to receive the result of the modal execution. You will then use that information to either progress the user or conduct appropriate error handling.

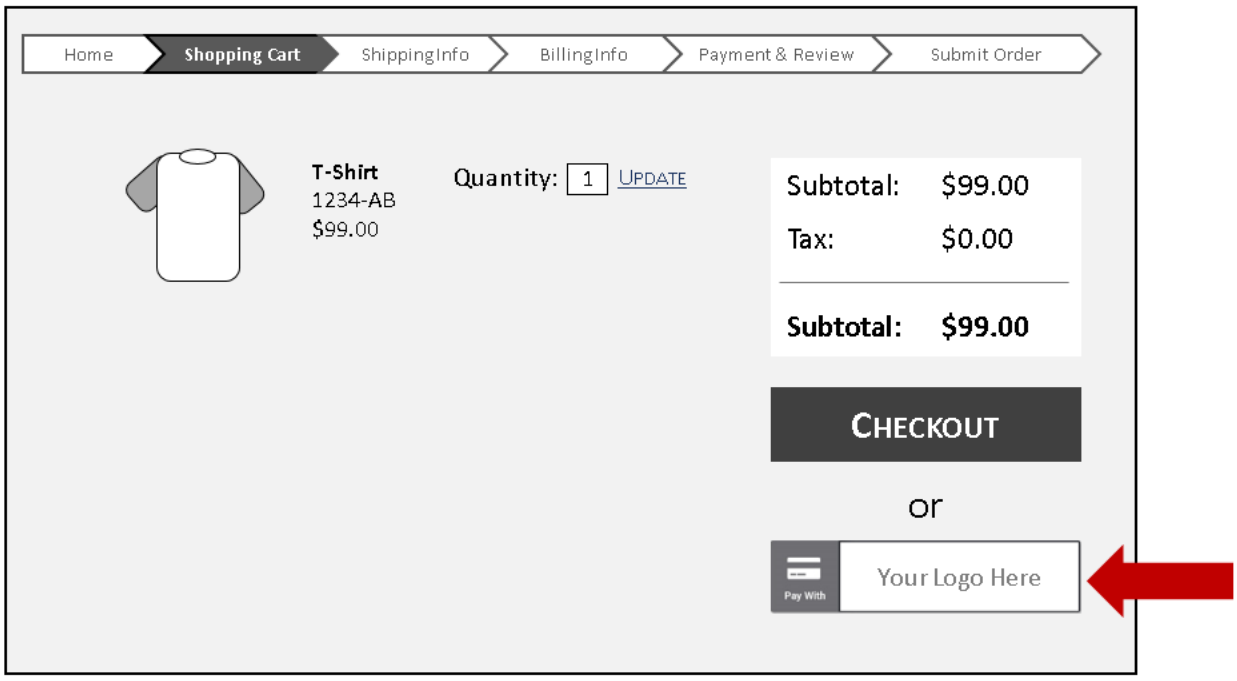
Each of these areas will be explored in greater detail within this guide.

3 Shopping Cart Setup

Before going too deeply into the technical details of how to implement Digital Buy, it is important to discuss the best practices for how the platform fits into the shopping cart. This section will discuss, at a high-level, optimal usage for a typical implementation. Specific implementations may have different needs. For special cases, it is recommended that a discussion occur to determine the best path forward.

3.1 Cart Level Button

Digital Buy has been designed to take advantage of placing a “Pay With” button at the cart subtotal level.



As an example, in the picture above, the customer is given the opportunity to make an immediate decision to pay with the private label or CareCredit card once they enter the cart, instead of deferring that decision to the tender selection stage of the checkout. By placing the “Pay With” button at this point in the customer’s checkout experience, several goals are achieved:

1) Data has shown that inclusion of a “rapid checkout” option (making a ‘Pay With’ button available at the cart level) has increased customer’s likelihood of completing the sale and the average ticket sizes of customers using rapid credit is larger than those that do not.
2) As a marketing opportunity, customers that may not be aware that financing is available will be made aware with the presence of the cart level button
3) Customers that click the “Pay With” button will have their billing information returned to the website, saving them the time to input their address manually
4) Customers that do not yet have a SYF private label or CareCredit account will have the opportunity to apply for credit from within the cart, and in the case of approvals (decisions are rendered in seconds), continue through checkout seamlessly without having to enter any additional information.

Synchrony will provide a custom button made specifically for each merchant’s unique needs, which should appear at the cart level. For CareCredit merchants, a CareCredit button will be provided.

In addition to the button appearing at the cart level, the SYF private label or CareCredit card should be the first option on the tender select screen.

3.2 Special Financing

One of the main value propositions for use of SYF private label and CareCredit cards is the opportunity to offer special financing offers to customers. To make full use of this value, it is important that the strategy for offering special financing from within the shopping cart is well defined and compliant.

A special financing offer is one where the customer is given special terms for paying off their purchase.

Did you know?
Generally, special financing through Synchrony Financial is offered in one of three formats:
Deferred Interest: Purchases made with deferred interest promotions offer terms where the customer is given a period of time (often referred to as the promotional period) to pay off the balance in full, after which time they will be responsible for all interest accrued to that point and in the future until it is paid. Interest begins accruing at the time of sale, and is not assessed if the balance is paid in full before the promotional period ends.
Reduced Interest: Reduced interest promotions offer a fixed payment schedule with a reduced APR rate over a specific rate of time.
No Interest: No interest promotions will never assess interest on the purchase, and are paid over a period of time with equal monthly payments.
It is recommended that you refer to your operating guide or contact your support team should you have questions on special financing offers.

All special financing offers are tied to a unique promotional code that indicates which offer should be applied to the corresponding amount in the purchase. As Digital Buy requires a promotional code to correspond to the total amount for the purchase, it is necessary for the shopping cart to have determined which promotion to apply prior to the last step of the checkout process being completed (the Order Completion Modal).

Commonly used strategies for special financing include (but are not limited to):

Single Offer: All orders receive the same promotion

Amount Based: Orders receive different financing offers (or "promotions") based on the amount being purchased. As an example, orders for <$500 may receive a 6-month deferred

	interest promotion, and orders for >=$500 receive a 12-month deferred interest promotion.
Product Based:	Orders receive specific promotions based on what products are being purchased. Please note that the option to use the card must be present for all products, even if a special financing offer is not offered for those products. *This option is not recommended if a simple integration is desired.*
Consumer's Choice:	The customer is given a list of promotions that their order has qualified for, and select the promotion that they wish to take advantage of. *This option is not recommended if a simple integration is desired.*

Ultimately, strategies for handling special financing offers are the responsibility of the merchant's (or provider's) site. It will be necessary for the merchant site to determine what special financing offers are to be utilized for each checkout, and to store the corresponding three-digit promotional codes for those offers. When executing the checkout, the specific promotional code (a three-digit numeric field) must be sent with the corresponding amount to apply to the purchase. It is recommended that you discuss this decision with your primary point-of-contact at Synchrony if you have questions.

3.3 Settlement/Funding

Credit card transactions generally have two stages that are necessary to complete. The first step is generally called "Authorization", which is where the decision is made on whether a consumer's account is in sufficient standing to make a purchase with a merchant (or provider) for a specific amount. When successful, the customer's

open-to-buy (credit limit minus outstanding balances/charges) is adjusted appropriately to tie up the funds until the transaction can be settled.

The second step is generally referred to as ‘Settlement’. The previously authorized transaction is ‘settled’, which at a high-level means that the merchant is paid, the customer’s balance is notated and the billing process begins.

Digital Buy can be set up to run in one of two modes. In the first mode, referred to as “Purchase” mode, all orders are set to automatically settle at the end of the day so the merchant does not have to concern themselves with processing the settlement step of the process. In this mode, SYF performs a host capture of each Purchase transaction and automatically sends the transactions to SYF systems for settlement processing each day.

The second mode supported by Digital Buy is “Authorize” mode. In this mode, all orders are authorized, but no settlement will take place until the merchant settles through a different channel.

Generally, it is recommended by SYF that you ask to use Digital Buy in “Purchase” mode unless there is a business need to authorize on the day the order is made online, then settle at a later point (usually when shipping may be delayed for more than a few days), it is recommended that you ask to use Digital Buy in “Purchase” mode.

If you do have a business need to run Digital Buy in “Authorize” mode, please talk to your primary point-of-contact at Synchrony.

4 Implementation

This section of the guide will detail the technical information necessary to integrate Digital Buy into the shopping cart, from start to finish. By implementing Digital Buy into the shopping cart, you agree to the Synchrony Bank License Terms for the Digital Buy Software that can be linked to from the Table of Contents.

4.1 Integration Checklist

Integration of Digital Buy requires that several tasks are completed prior to being certified to go live. The checklist below is a high-level summary of those tasks, from start to finish.

Integration Checklist
☑ Integration documentation obtained
☑ IT deliverables furnished to SYF (domain, post back URLs)
☑ Marketing deliverables provided to SYF
☑ Certification scripts obtained
☑ UAT region credentials obtained
☑ JavaScript libraries included on website
☑ Rapid checkout (“Pay With”) button obtained from SYF
☑ Rapid checkout button added to shopping cart (Authenticate, Consumer Information Modal)
☑ Cart level authentication response handling
☑ Cart level consumer information modal response handling
☑ Tender type added to tender selection
☑ Complete order, cart button pressed (order completion modal)
☑ Complete order, cart button pressed - order completion modal response handling

- ☑ Complete order, cart button not pressed (authenticate, combined modal)
- ☑ Complete order, cart button not pressed – combined modal response handling
- ☑ Certification test cases executed
- ☑ Certification passed
- ☑ Production Credentials Received

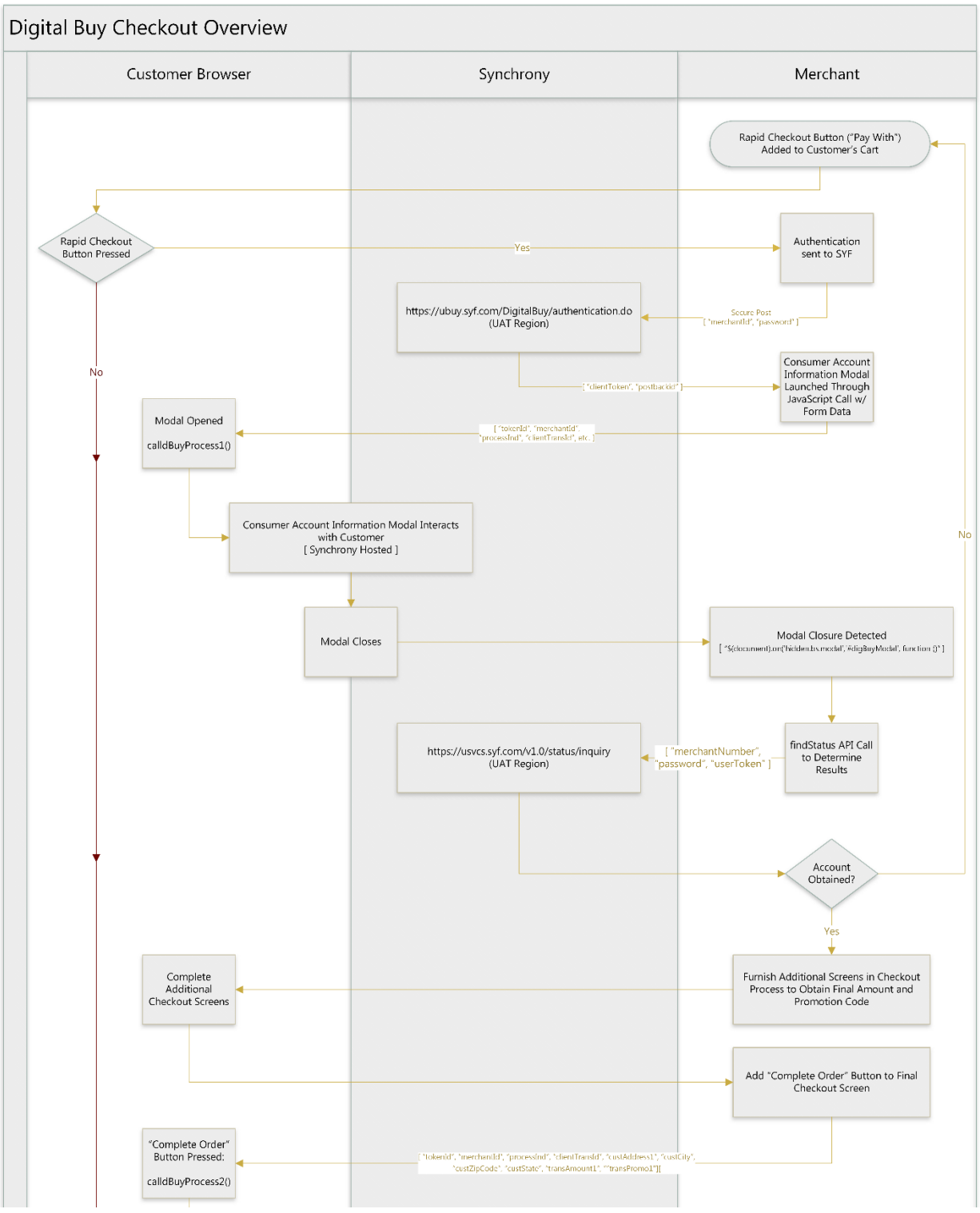
4.2 Website Setup

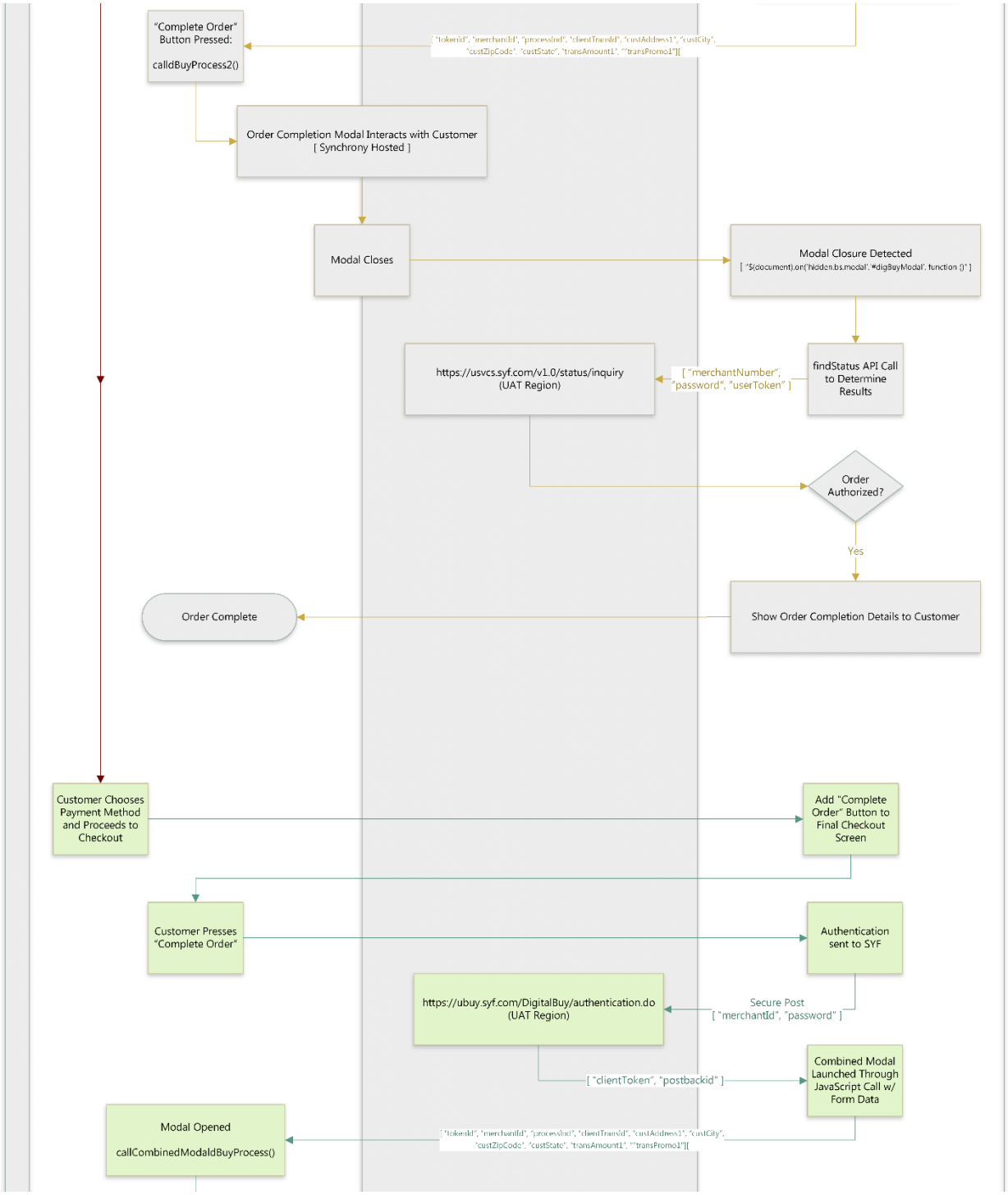
As the end-to-end checkout process requires an interplay between the shopping cart and Digital Buy, there are certain pre-requisite activities that must be completed before a transaction can be successful.

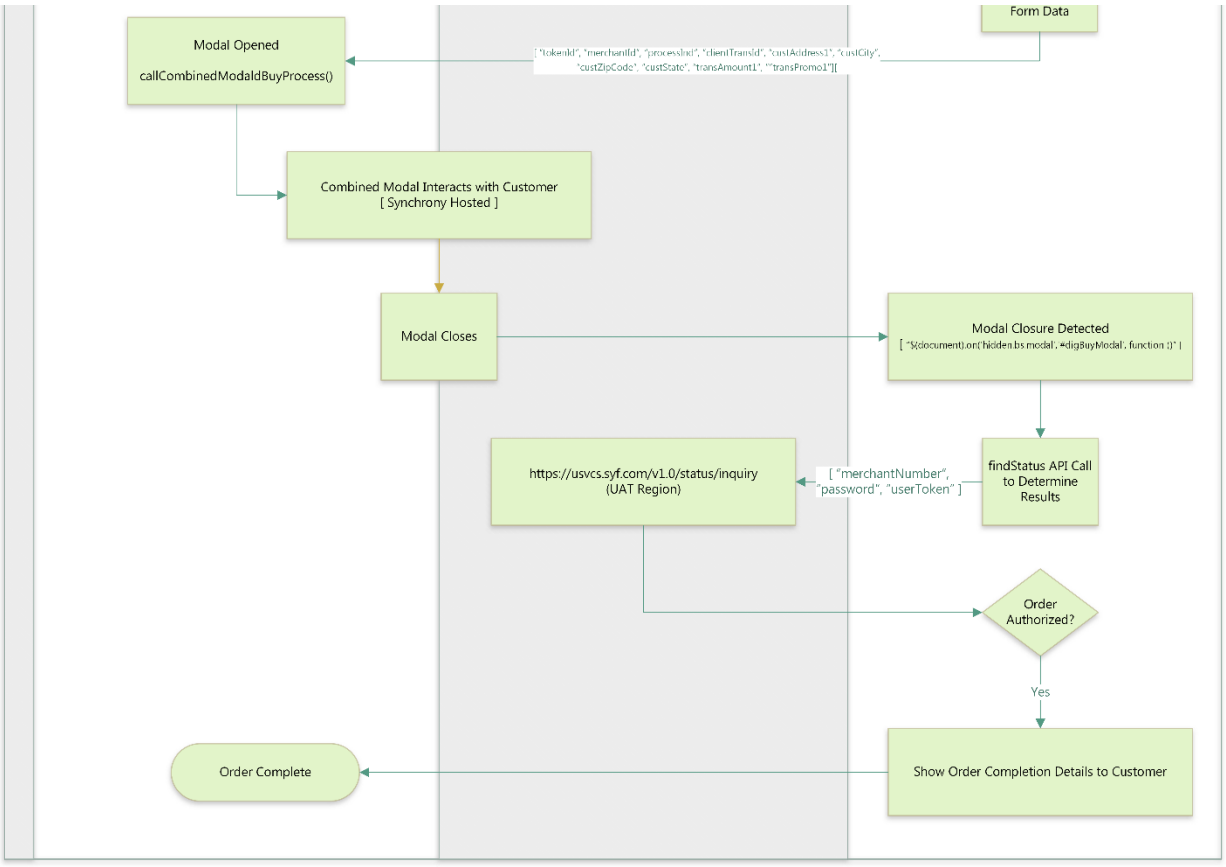
4.2.1 Recommended Checkout Flow

A high-level flow chart for the two different possible checkout scenarios (whether the customer chooses to click the rapid checkout “Pay With” button or not) is provided on the following pages. Details on the steps shown in the flow are found throughout the rest of the guide.

You may choose to receive status updates slightly differently based on your preference or what might be easier for your site.


Digital Buy Checkout Overview
Customer Browser
Synchrony
Merchant
Rapid Checkout Button ("Pay With") Added to Customer's Cart
Rapid Checkout Button Pressed
Yes
Authentication sent to SYF
https://ubuy.syf.com/DigitalBuy/authentication.do (UAT Region)
Secure Post ["merchantId", "password"]
["clientToken", "postbackId"]
Consumer Account Information Modal Launched Through JavaScript Call w/ Form Data
No
Modal Opened calldBuyProcess1()
["tokenId", "merchantId", "processInd", "clientTransId", etc.]
Consumer Account Information Modal Interacts with Customer [Synchrony Hosted]
Modal Closes
Modal Closure Detected ["$(document).on('hidden.bs.modal', '#digBuyModal', function () }"]
No
https://usvcs.syf.com/v1.0/status/inquiry (UAT Region)
["merchantNumber", "password", "userToken"]
findStatus API Call to Determine Results
Account Obtained?
Yes
Complete Additional Checkout Screens
Furnish Additional Screens in Checkout Process to Obtain Final Amount and Promotion Code
Add "Complete Order" Button to Final Checkout Screen
"Complete Order" Button Pressed: calldBuyProcess2()
["tokenId", "merchantId", "processInd", "clientTransId", "custAddress1", "custCity", "custZipCode", "custState", "transAmountL", "transPromo1"][


"Complete Order" Button Pressed:
callBuyProcess2()
["tokenId", "merchantId", "processId", "clientTransId", "custAddress1", "custCity", "custZipCode", "custState", "transAmount1", ""transPromo1"][
Order Completion Modal Interacts with Customer [Synchrony Hosted]
Modal Closes
Modal Closure Detected
["$(document).on('hidden.bs.modal', '#digBuyModal', function () {"]
https://usvcs.syf.com/v1.0/status/inquiry (UAT Region)
["merchantNumber", "password", "userToken"]
findStatus API Call to Determine Results
Order Authorized?
Yes
Order Complete
Show Order Completion Details to Customer
Customer Chooses Payment Method and Proceeds to Checkout
Add "Complete Order" Button to Final Checkout Screen
Customer Presses "Complete Order"
Authentication sent to SYF
https://ubuy.syf.com/DigitalBuy/authentication.do (UAT Region)
Secure Post
["merchantId", "password"]
["clientToken", "postbackid"]
Combined Modal Launched Through JavaScript Call w/ Form Data
Modal Opened
callCombinedModaldBuyProcess()
["tokenId", "merchantId", "processId", "clientTransId", "custAddress1", "custCity", "custZipCode", "custState", "transAmount1", ""transPromo1"][


Form Data
Modal Opened
callCombinedModalBuyProcess()
["tokenId", "merchantId", "processId", "clientTransId", "custAddress1", "custCity",
"custZipCode", "custState", "transAmount1", "transPromo1"][
Combined Modal Interacts with Customer
[Synchrony Hosted]
Modal Closes
Modal Closure Detected
["$(document).on('hidden.bs.modal', '#digBuyModal', function () {"]
findStatus API Call
to Determine
Results
["merchantNumber",
"password", "userToken"]
https://usvcs.syf.com/v1.0/status/inquiry
(UAT Region)
Order
Authorized?
Yes
Show Order Completion Details to Customer
Order Complete

4.2.2 Digital Buy Source Files

The following files must be included on the site for Digital Buy to function.

Integration Region	
DB Bootstrap Libraries:	https://ubuy.syf.com/digitalbuy/js/bootstrap.min.js
DB Modal Libraries:	https://ubuy.syf.com/digitalbuy/js/jquery-1.9.1.min.js
DB Script Support:	https://ubuy.syf.com/digitalbuy/js/merchant.js
CSS Files:	https://ubuy.syf.com/digitalbuy/css/ bootstrap.min.css

Production Region	
DB Bootstrap Libraries:	https://buy.syf.com/digitalbuy/js/bootstrap.min.js
DB Modal Libraries:	https://buy.syf.com/digitalbuy/js/jquery-1.9.1.min.js
DB Script Support:	https://buy.syf.com/digitalbuy/js/merchant.js
CSS Files:	https://buy.syf.com/digitalbuy/css/ bootstrap.min.css

An example of how to include these files is given below:

Digital Buy Included Files Example

```
<script src="https://ubuy.syf.com/digitalbuy/js/bootstrap.min.js"></script>
<script src="https://ubuy.syf.com/digitalbuy/js/jquery-1.9.1.min.js"> </script>
<script src=" https://ubuy.syf.com/digitalbuy/js/merchant.js"></script>
<link href="https://ubuy.syf.com/digitalbuy/css/bootstrap.min.css" rel="stylesheet">
```

4.3 Authentication

During each checkout experience, it is necessary to obtain fresh tokens (unique, checkout specific identifiers) from the Digital Buy platform. The process used to obtain these tokens is referred to as the "Authentication" step.

Authentication works in a straightforward manner. Whenever a customer is ready to checkout with the Digital Buy platform, the shopping cart should perform a secure POST method to one of the two following URLs, depending on whether the request is to the test or production region:

Test Region: https://ubuy.syf.com/DigitalBuy/authentication.do
Production Region: https://buy.syf.com/DigitalBuy/authentication.do

The body of the secure POST authentication request should contain two name-value pairs:

- "**merchantId**" and the corresponding Merchant ID (MID) provided by SYF
- "**password**" and the corresponding password provided by SYF

These parameters represent the merchant's (or provider's) unique credentials to access the Digital Buy platform. *They should never be exposed to the public, and any code containing the cleartext values must be properly secured.*

Please note: The authentication request parameters must be sent as a server-side secure form post, not as a standard browser call.

Upon successful receipt of valid credentials (the two name-value pairs listed above), SYF will respond to the POST with a pair of tokens in JavaScript Object Notation

(JSON) format. The specific name-value pairs returned in the Authentication response are:

- "**clientToken**" and the corresponding token that will be used to execute the consumer-facing modal methods. This token will be exposed to the customer's browser and should not be considered secure. The "clientToken" parameter has a length of 29 characters.
- "**postbackid**" and the corresponding token that will be used to validate that the final authorization response came from SYF rather than a third party. *This 'postbackid' token should never be exposed to the consumer* (or the consumer's browser), and instead must be stored securely. The "postbackid" parameter has a maximum length of 50 characters.

Once the tokens are obtained, the checkout experience can continue by calling the first modal (Consumer Account Information modal).

Things to Remember…
Tokens will expire after ten minutes if the consumer enters their account number directly or uses the option to find their account number through an account lookup. If the consumer applies for credit, tokens will expire after fifteen minutes. **If a token expired response is received at any point, it will be necessary to start that checkout process over with a new authentication and set of tokens.**

Sample code for the Authentication step is given below in both Java and PHP. This code was written with the assumption that there is a proxy in place for outbound connections. If a proxy is not in use, then this code sample will not function as written.

Authentication Example – Java

```
// The URL below is to target the test region
String url="https://ubuy.syf.com/DigitalBuy/authentication.do";
```

```
// The URL below is to target the production region (currently commented out)
//String url="https://buy.syf.com/DigitalBuy/authentication.do";

SSLContext sslcontext = SSLContexts.createSystemDefault();
String[] arr={"TLSv1.1","TLSv1.2"}; // Supported TLS Protocol versions

SSLConnectionSocketFactory sslConnectionSocketFactory = new
SSLConnectionSocketFactory(sslcontext, arr, null,
SSLConnectionSocketFactory.getDefaultHostnameVerifier());

// Replace <proxy hostname> and <proxy prot> with the appropriate values
HttpHost proxy = new HttpHost(System.getProperty("<proxy hostname>"), <proxy
port>);

DefaultProxyRoutePlanner routePlanner = new DefaultProxyRoutePlanner(proxy);
CloseableHttpClient httpClient =
HttpClients.custom().setRoutePlanner(routePlanner).build();
httpClient =
HttpClients.custom().setSSLSocketFactory(sslConnectionSocketFactory).setProxy(pr
oxy).build();
HttpPost httpPost = new HttpPost(url);
httpPost.addHeader("User-Agent", "Mozilla/5.0");

if (url != null)
{
   try
   {
      List<NameValuePair> urlParameters = new ArrayList<NameValuePair>();

      // Replace <MerchantID> with MID
      urlParameters.add(new BasicNameValuePair("merchantId",<MerchantID>));

      // Replace <password> with password
      urlParameters.add(new BasicNameValuePair("password",<password>));
```

```
        HttpEntity postParams = new UrlEncodedFormEntity(urlParameters);
        httpPost.setEntity(postParams);

        CloseableHttpResponse httpResponse = httpClient.execute(httpPost);

        int statusCode = httpResponse.getStatusLine().getStatusCode();

        try
        {
           jsonString = EntityUtils.toString(httpResponse.getEntity());
           JSONObject obj = new JSONObject(jsonString);
           token = obj.getString("clientToken");
           postbackid =obj.getString("postbackid");
           token = token.replaceAll("[\\\t|\\\n|\\\r]","");
        } finally {
           httpResponse.close();
        } catch (UnsupportedEncodingException e) {
                 e.printStackTrace();
        } catch (ClientProtocolException e) {
                 e.printStackTrace();
        } catch (IOException e) {
                 e.printStackTrace();
        }
}
```

Authentication Example – PHP

```
<?php

$ch = curl_init();
$config_method = 'POST';

// Replace 'MXXX' with MID
// Replace 'PXXX' with password
$fields = array("merchantId" => "MXXX", "password" => "PXXX");
$config_headers[] = 'Accept: application/xml';
```

```
// The URL below is to target the test region
$config_address = "https://ubuy.syf.com/DigitalBuy/authentication.do";

// The URL below is to target the production region (currently commented out)
//$config_address = ' https://buy.syf.com/DigitalBuy/authentication.do';

$chandle = curl_init();

curl_setopt($chandle, CURLOPT_RETURNTRANSFER, true);
curl_setopt($chandle, CURLOPT_URL, $config_address);
curl_setopt($chandle, CURLOPT_HTTPHEADER, $config_headers);
curl_setopt($chandle, CURLOPT_POSTFIELDS, http_build_query($fields));
curl_setopt($chandle, CURLOPT_SSL_VERIFYPEER, false);

$response = curl_exec($chandle);
$errmsg = curl_error($chandle);
$cInfo = curl_getinfo($chandle);
curl_close($chandle);

if ($errmsg =="")
{
   $responseobj=json_decode($response);
   if($responseobj != null)
   {
      echo "postbackid:". $responseobj->postbackid ."<br>";
      echo "clientToken:" . $responseobj->clientToken ."<br>";
   }
   else
   {
      echo "<b style='color:red;'>Tokens Not Generated Successfully</b><br
/><pre>";
   }
}
Else
{
```

```
    echo "<b style='color:red;'>There is a Problem, ". $errmsg ."</b>";
}

?>
```

4.4 Consumer Account Information Modal

The consumer account information modal is the first of two modals used by Digital Buy during the consumer's checkout experience. The purpose of this modal is to obtain the consumer's account number through either direct account number entry, account number lookup, or through a new credit application (for new customers).

Did you know?

"Modal" refers to a modal window. A modal window is executed by disabling the current window and opening a new, typically smaller, window on top of it, with the original window greyed out (but still visible). This allows for the look-and-feel that the original window is still there and leads to a less disconnected user experience. When the modal window's actions are complete, it closes and the original window reactivates.

Digital Buy uses the standard open-source Twitter Bootstrap version 3 to execute two modal windows. The Twitter Bootstrap libraries are used extensively across the web, and there is significant documentation publicly available should there be a need for detailed information (such as the need to hook specific triggers).

4.4.1 Consumer Account Information Modal Execution

The Consumer Account Information Modal is opened by calling the "calldBuyProcess1" function (included in the Digital Buy library files) with an argument of a hidden HTML form containing the parameters to be passed to the modal window. It must be called from within a element with a tag id of "dbuymodel1".

At a bare minimum, parameters that must be included are the client token ID ("tokenId"), the merchant identification number ("merchantID"), the process

indicator (“processInd”), and the client generated transaction ID (“clientTransId”). The full parameter list is shown below.

Parameter	Type	Data Length	Required	Comments
tokenId	String	29	Yes	This parameter correlates to the “clientToken” returned in the authenticate request. This value uniquely identifies the specific checkout.
merchantID	String	16	Yes	This is the unique identifier assigned by SYF to identify the specific merchant.
processInd	Integer	1	Yes	This parameter must be passed as “1” to indicate that it is the Consumer Account Information modal.
clientTransId	String	30 Max	Yes	This is a merchant generated unique identifier that will be echoed back in the modal responses. This parameter must be unique, and must be passed as it will be the only way that responses can be isolated in the case of expired or bad tokens.
custFirstName	String	20	No	When present, this parameter will be used to prefill the customer’s first name for account lookup and credit application entry. Alphabetic only.
custLastName	String	25	No	When present, this parameter will be used to prefill the customer’s first name for account lookup and credit application entry. Alphabetic only.
custZipCode	String	5 or 9	No	When present, this parameter will be used to prefill the customer’s ZIP Code.
cardNumber	String	16	No	When present, this parameter will be used to prefill the customer’s SYF account number for direct account number entry.
iniPurAmt	String	6	No	When present, this parameter will be used to prefill the initial purchase amount for credit application entry.
custAddress1	String	25	No	When present, this parameter will be used to prefill the first line of the

				customer's address for credit application entry.
custAddress2	String	25	No	When present, this parameter will be used to prefill the second line of the customer's address for credit application entry.
phoneNumber	String	10	No	When present, this parameter will be used to prefill the customer's phone number for credit application entry.
emailAddress	String	60	No	When present, this parameter will be used to prefill the customer's email address for credit application entry.
custCity	String	20	No	When present, this parameter will be used to prefill the customer's city for credit application entry.
custState	String	2	No	When present, this parameter will be used to prefill the customer's state for credit application entry.

Consumer Account Information Modal Window Example

```
<div id="dbuymodel1"/>
<form name="dbuyform1" id="dbuyform1">
                    <input type="hidden" name="processInd" value="1"/>
                    <input type="hidden" name="tokenId" value="< tokenId>"/>
                    <input type="hidden" name="merchantID" value="< merchantID >"/>
                    <input type="hidden" name="clientTransId" value="< clientTransId >"/>
                    <input type="hidden" name="custFirstName" value="< custFirstName >"/>
                    <input type="hidden" name="custLastName" value="< custLastName >"/>
                    <input type="hidden" name="custZipCode" value=" <custZipCode>"/>
                    <input type="hidden" name="cardNumber" value="< cardNumber >"/>
<button type="button" data-toggle="modal" data-target="#digBuyModal"
onclick="calldBuyProcess1(this.form)">Checkout</button>/
</form>
```

4.4.2 Consumer Account Information Modal Response Handling

Once the consumer completes their activity in the Consumer Account Information Modal, that modal will close and a call should be made to the Status Inquiry API and/or a response will be sent via secure POST to the listener URL (“postback URL”).

Please note that in cases where the consumer has chosen to apply for credit, a credit application will open in a separate tab, but the modal will remain open on the merchant (or provider) website and display a message stating “Credit application in process, please complete the application in the other browser window” and an option to close the modal should they decide not appply.

4.4.2.1 Modal Closure Detection

Once the modal window closes, should you choose the Status Inquriy API for updates, an immediate call should be made to the API. If using the listener URL, the result should be posted to the listener URL within milliseconds. It is therefore recommended that the page serving the customer detect the closure of the modal window, then begin checking for the presence of the post back response.

Modal Window Closure Detection Example Code Wrapper

```
$(document).on('hidden.bs.modal','#digBuyModal', function () {
    // Make call to findStatus to obtain full response
});
```

4.4.2.2 Response Format

The result of the Consumer Account Information modal can be obtained with the Status Inquiry API call and/or posted to the listener URL in JSON format with the following elements:

Parameter	Type	Data Length	Comments	Sent in Callback	Sent in API Call \Postback
TokenId	String	29	This parameter correlates to the "clientToken" returned in the authenticate request. This value uniquely identifies the specific checkout.	Yes	Yes
StatusCode	Integer	2 or 3	This vaue identifies the result of the attempt to get the consumer's account information in the Consumer Account Information Modal.	Yes	Yes
StatusMessage	String	51 Max	This value is a string representation of the result indicated by StatusCode above.	Yes	Yes
AccountNumber	String	16 Max	This parameter returns the consumer's account number in a masked format (last 4 of account showing, the remainder is masked).	Yes	Yes
FirstName	String	20 Max	When an account is successfully obtained, this parameter contains the cardholder's first name.	Yes	Yes
LastName	String	25 Max	When an account is successfully obtained, this parameter contains the cardholder's last name.	Yes	Yes
Address1	String	25 Max	When an account is successfully obtained, this parameter contains the cardholder's address (line 1).	Yes	Yes
Address2	String	25 Max	When an account is successfully obtained, this parameter contains the cardholder's address (line 2).	Yes	Yes

City	String	20 Max	When an account is successfully obtained, this parameter contains the cardholder's city.	Yes	Yes
State	String	2 Max	When an account is successfully obtained, this parameter contains the cardholder's state.	Yes	Yes
zipCode	String	5 or 9 Max	When an account is successfully obtained, this parameter contains the cardholder's ZIP code.	Yes	Yes

Status Code	Status Message
400	Required input not provided/Input validation error (Token not found in DB)
401	Transaction Failed (Token Expired Scenario)
100	Customer Terminated
002	Account Authentication success (Customer gets the Account Authentication success confirmation through ANL scenario)
003	Account Authentication success (Customer gets the Account Authentication success confirmation through EAN scenario)
010	Customer is applying for credit (second post will be sent when application process is completed)
00, also possible (21, 22, 24, 25, 26, 27, 28)	eApply_Transaction_Approved
04, also possible (06, 99)	eApply_Transaction_Pending
07, also possible (08, 12, 13, 14, 15, 16, 17, 18, 81, 82, 90)	eApply_Transaction_Declined
09 or NO Code Sent	eApply_CALL_NOW
03, 10, 11	eApply_PROCESS_ERROR
500	System Exception

4.4.2.3 Account Authentication Success Scenario

When a customer's account information is successfully obtained with the Consumer Account Information modal window, the modal will close and a response with a status code of '002', '003', or '00' will be returned to the Status Inquiry API call and/or sent to the Listener URL depending on the option you choose for your sites integration. For each of these cases, the customer can be advanced forward in the checkout process.

4.4.2.4 Apply Scenario

When a customer does not already have an account, they may choose to apply for one from the Account Information modal. In these cases, a credit application will open in a separate tab, and the modal will remain open on the merchant (or provider) website with a message stating "Credit application in process, please complete the application in the other browser window" and an option to close the modal should they decide not apply.

An example of what is displayed within the modal when the customer chooses to apply is shown below.

4.4.2.4.1 Apply Scenario – Approved

Once that customer completes the application process, and is approved for a new account, the data returned via Callback Message or findStatus API call will contain a status code of '**00'**. While rare, it is also possible to receive 21, 22, 24, 25, 26, 27, or 28 in the application approval scenario. When this result is received, the customer can be advanced to the next stage of the checkout process, as an account number has been successfully obtained for the customer.

4.4.2.4.2 Apply Scenario – Declined

For scenarios where the customer is declined for a new account, the data returned via Callback Message or findStatus API will contain a status code of '**07'**. While rare, it is also possible to receive 08, 12, 13, 14, 15, 16, 17, 18, 81, 82, or 90 in the application declined scenario. In these cases, it should be treated as a failure to obtain an account number, and the user returned to the tender selection portion of the checkout process.

4.4.2.4.3 Apply Scenario - Pending

On rare occasions, there will be a scenario where Synchrony will be unable to reach an immediate decision on the customer's credit application (an example of this would be if the customer has a hold on their credit bureau). When this occurs, the credit application will be placed into a 'pending' state and the data returned via Callback Message or findStatus API call will contain a status code of '**04'**. While rare, it is also possible to receive 06 or 99 in the application pending scenario. When these scenarios occur, they must be treated as a failure to obtain an account number, and the user returned to the tender selection portion of the checkout process.

4.4.2.4.4 Apply Scenario – Exception Responses

On very rare occasions, there could be a scenario where an internal systems issue might result in a situation where an application was not processed for the customer. When this occurs, the credit application will be placed into a 'pending' state and the data returned via Callback Message or findStatus API call will contain a status code of '03', '09' '10' or '11'(*see table for status message). Very rarely, an empty '' status code with the Status Message of 'eApply_CALL_NOW' can be sent. This should only occur during an outage situation. When these scenarios occur, they must be treated as a failure to obtain an account number, and the user returned to the tender selection portion of the checkout process.

4.4.2.5 Account Authentication Failure Scenario

If Digital Buy is not able to successfully obtain an account number for the customer with the Consumer Account Information Modal, then it will return one of several responses that can all be treated as a failure scenario. These cases include '400' (input validation error or token not found in database), '401' (token expired), and '100' (user terminated modal without providing an account number). For each of these cases, the user should be given an error message and returned to tender type selection.

4.4.2.6 Account Information Exception Handling

In some cases, Digital Buy will return a status code of '500' to the Callback Message or Status Inquiry API. This indicates that there was a system error. This error should be logged, and if occurring regularly, escalated for resolution as intervention may be necessary to restore functionality. When these scenarios do occur, the user experience should be handled in the same way as an account authentication failure. An error message should be shown, and the user returned to tender type selection in checkout.

4.4.2.7 Account Information Timeout Error Handling

Rarely, a response may not be received by the Callback Message or Status Inquiry API. The cause for this could be due to network failures or configuration settings, but it should generally be an edge case. Nevertheless, to ensure a smooth user experience, the web processes handling the individual user checkouts should have a timeout period when waiting for a response.

While the specific timeout period can vary based on business decisions, it is generally recommended that the timeout period to receive and process the first postback message be set to no longer than 30 seconds after the Consumer Account Information Modal closes.

4.4.3 Returned Customer Address Information

When the Consumer Account Information Modal successfully obtains an account number for the customer, the customer's address on file is included in the response message sent to the Callback Message or Status Inquiry API call. This information should be used to pre-fill the customer's billing address if one does not already exist.

It is important to note that if an address that does not match this address is sent to the Order Completion modal, then the transaction will not authorize due to an address mismatch.

If the decision is made to pass a different address to Digital Buy than the address where the product is being shipped (in order to bypass this check), it is highly recommended that additional steps be taken to Know Your Customer before conducting business with them. If this route is intended, potential responsibility for fraudulent activity should be discussed with your primary Synchrony point-of-contact.

4.5 Order Completion Modal

The Order Completion Modal is the second of the two Digital Buy modals that are executed during the checkout process. The purpose of this modal is to complete the transaction by authorizing the account for the purchase amount and corresponding promotional codes. If necessary, the modal will also initiate a credit line increase request for customers that do not have sufficient open-to-buy to complete the purchase.

4.5.1 Order Completion Modal Execution

The Order Completion Modal is opened by calling the “calldBuyProcess2” function (included in the Digital Buy library files) with an argument of a hidden HTML form containing the parameters to be passed to the modal window. It must be called from within a element with a tag id of “dbuymodel2”.

At a bare minimum, parameters that must be included are the client token ID (“tokenId”), the merchant identification number (“merchantID”), the process indicator (“processInd”), the customer address fields (“custAddress1”, custAddress2”, “custCity”, “custState”, and “custZipCode”), the promotional code to apply to the purchase (“transPromo1”), and the amount of the purchase (“transAmount1”). The full parameter list is shown below.

Parameter	Type	Data Length	Required	Comments
TokenId	String	29	Yes	This parameter correlates to the “clientToken” returned in the authenticate request. This value uniquely identifies the specific checkout.
merchantID	String	16	Yes	This is the unique identifier assigned by SYF to identify the specific merchant.
processInd	Integer	1	Yes	This parameter must be passed as “2” to indicate that it is the Order Completion modal .

custFirstName	String	20	No	The customer's first name, alphabetic only.
custLastName	String	25	No	The customer's last name, alphabetic only.
custAddress1	String	25	Yes	The first line of the customer's address, alphanumeric.
custAddress2	String	25	No	The second line of the customer's address, alphanumeric.
custCity	String	20	Yes	The customer's city, alphanumeric.
custState	String	2	Yes	The customer's state abbreviation, alphabetic only.
custZipCode	String	5 or 9	Yes	The customer's ZIP Code, numeric only.
transPromo1	String	3	Yes	The three-digit numeric "promo code" that indicates what special financing offer should be applied to "transAmount1". This number should be provided by the Relationship Manager if not known.
transAmount1	BigDecimal	6.2	Yes	The amount to be authorized under the promotion passed in "transPromo1". Must include both decimal places and amounts up to 999999.99 are accepted.
transPromo2	String	3	Cond.	Optional field that should only be sent if multiple finance offers are in use for the same transaction. The three-digit numeric promotional code to apply to "transAmount2". Must be included in if "transAmount2" is included.
transAmount2	BigDecimal	6.2	Cond.	Optional field that should only be sent if multiple finance offers are in use for the same transaction. The amount to be authorized under the promotion passed in "transPromo1". Must include both decimal places and amounts up to 999999.99 are accepted.

transPromo3	String	3	Cond.	Optional field that should only be sent if multiple finance offers are in use for the same transaction. The three-digit numeric promotional code to apply to “transAmount3”. Must be included in if “transAmount2” is included.
transAmount3	BigDecimal	6.2	Cond.	Optional field that should only be sent if multiple finance offers are in use for the same transaction. The amount to be authorized under the promotion passed in “transPromo3”. Must include both decimal places and amounts up to 999999.99 are accepted.
defaultPromoCode	String	3	Cond.	Optional field that should only be sent if multiple special financing offers are being applied in the same transaction. This three-digit numeric field indicates the default promotion that should be applied for authorization purposes. The Relationship Manager will provide this value if used.

Order Completion Modal Window Example

```
<div id="dbuymodel2"/>

<form name="dbuyform2" id="dbuyform2">
                        <input type="hidden" name="processInd" value="2"/>
                        <input type="hidden" name="tokenId" value="<tokenId>"/>
                        <input type="hidden" name="merchantID" value="<merchantID>"/>
                        <input type="hidden" name="custFirstName" value="<custFirstName>"/>
                        <input type="hidden" name="custLastName" value="<custLastName>"/>
                        <input type="hidden" name="cardNumber" value="<cardNumber>"/>
                        <input type="hidden" name="custAddress1" value="<custAddress1>"/>
                        <input type="hidden" name="custAddress2" value="<custAddress2>"/>
                        <input type="hidden" name="custCity" value="<custCity>"/>
                        <input type="hidden" name="custState" value="<custState>"/>
                        <input type="hidden" name="custZipCode" value="<custZipCode>"/>
                        <input type="hidden" name="transPromo1" value="<transPromo1>"/>
                        <input type="hidden" name="transAmount1" value="<transAmount1>"/>
                        <input type="hidden" name="transPromo2" value="<transPromo2>"/>
                        <input type="hidden" name="transAmount2" value="<transAmount2>"/>
                        <input type="hidden" name="transPromo3" value="<transPromo3>"/>
                        <input type="hidden" name="transAmount3" value="<transAmount3>"/>
                        <input type="text" name="defaultPromoCode" value="<defaultPromoCode>"/>
```

```
<button type="button" data-toggle="modal" data-target="#digBuyModal"
onclick="calldBuyProcess2(this.form)">Checkout</button>
</form>
```

4.5.2 Order Completion Modal Response Handling

Once the consumer completes their activity in the Order Completion Modal, that modal will close and the results can be obtained via the Callback Message or the Status Inquiry API.

4.5.2.1 Order Completion Modal Response Format

The result of the Order Completion modal can be obtained via the Status Inquiry API call and/or will be posted to the listener URL in JSON format with the following elements:

Parameter	Type	Data Length	Comments	Sent in Callback	Sent in API Call \Postback
tokenId	String	29	This parameter correlates to the “clientToken” returned in the authenticate request. This value uniquely identifies the specific checkout.	Yes	Yes
Status Code	Integer	3	This vaue identifies the result of the attempt to complete the transaction in the Order Completion Modal.	Yes	Yes
Status Message	String	51 Max	This value is a string representation of the result indicated by StatusCode above.	Yes	Yes
ClientTransactionID	String	30 Max	This field echoes back the “clientTransId” sent in the Consumer Account Information Modal request.	Yes	Yes

TransactionAmount	String	6.2	This field provides the total amount for the transaction .	Yes	Yes
TransactionDate	String	30 Max	The date and timestamp of the transaction – Example: Mon Sep 12 00:45:33 PDT 2016 .	Yes	Yes
TransactionDescription	String	Max 13	This field is a string representation of whether the transaction was processed as an "AUTHORIZATION" or a "PURCHASE".	Yes	Yes
AuthCode	String	6 Max	This field contains the six-digit numeric authorization code for the transaction. Example: 013798.	Yes	Yes
accountNumber	String	16	All but the last 4 digits of the account number will be sent masked. **Via API Call and Postback (legacy) ONLY, if the Transaction is run in Authorization mode, and is successful, the full unmasked account number will be sent in this field for later use in settlement. *This field is not sent unmasked if tokenization is enabled.*	Yes – Always Masked – See comments	Yes – Full Acct Number in Auth Mode – See Comments
accountToken	String	24	This field returns the account token that can be used later to settle the transaction. It replaces the unmasked account number when used.	No	Yes
PromoCode	String	3 Max	This field echoes back the three-digit numeric code that was applied to "transAmount1", or the default promotional code for transactions with multiple promotions in use.	Yes	Yes
FirstName	String	20 Max	The customer's first name.	Yes	Yes
LastName	String	25 Max	The customer's last name.	Yes	Yes

PostbackId	String		This field contains the postbackid that is sent in the initial authentication response. It should be compared to the value stored security on the server to confirm that the response is from SYF and not a third party.	No	Yes

Status Code	Status Message
000	Authorization Approved/ Purchase Success
001	Authorization Declined/Purchase Declined
400	Required input not provided/Input validation error (Token not found in DB)
401	Transaction Failed (Token Expired Scenario)
402	Promo code validation fail
403	Address verification check Fail
100	Customer Terminated
500	System Exception

4.5.2.2 Authorization Successful Scenario

In cases where the Order Completion Modal successfully obtains an authorization, the Digital Buy platform will return a status code '000' to the Callback Message or Status Inquiry API. In these cases, the customer's purchase can be considered complete for the amount sent. When appropriate, the cart should be cleared and the customer allowed to continue with a fresh order.

If Digital Buy is running in Purchase mode, settlement will occur automatically overnight. If it is running in Authorization mode, then the full account number (or account token, when enabled) must be retrieved at the end of the transaction when the findStatus API is called. It is required for settlement later (through a different channel) by the merchant. Full account numbers (as well as account tokens) should be stored in a PCI compliant manner.

4.5.2.3 Authorization Failure Scenario

The Order Completion Modal step of the purchase process can fail for several different reasons. Each of these may potentially need to be handled differently depending on the specific checkout environment.

4.5.2.3.1 Authorization Declined

An Authorization Declined response (status code '001') occurs when the customer's account is not approved for the transaction as sent. In this scenario, it is necessary to return the user to tender select so they can choose an alternate tender type to complete the order.

4.5.2.3.2 Input Validation/Token Not Found

Status code '400' indicates that an invalid token was used (or one that is no longer present in the Digital Buy database), or the input provided was invalid. When this code is received, it should be logged and, if recurring escalated for resolution as it may indicate that action is needed to correct the issue.

4.5.2.3.3 Token Expired Scenario

Tokens expire 10 minutes (15 minutes when a credit application takes place) after the initial authentication occurs. In cases where the Order Completion Modal is called outside this expiration window, the modal will fail and Digital Buy will return a status code '401' to the Status Inquiry API and/or listener.

When this scenario occurs, the Digital Buy checkout process should be retried, including all steps (Authorization, the Consumer Account Information Modal, and the Order Completion modal).

4.5.2.3.4 Promotional Code Validation Error Scenario

When the three-digit numerical code that controls what special financing should be applied to the order is unable to be validated, a status code ‘402’ will be returned to the Callback Message or Status Inquiry API. This error indicates that either a promotional code has been configured incorrectly, or an existing promotional code has expired. These errors should be escalated to your primary point-of-contact at Synchrony, as they typically require manual intervention to resolve.

In this scenario, the customer should be returned to tender selection.

4.5.2.3.5 Address Verification Failure Scenario

If the customer address passed to the Order Completion does not match the address on the SYF systems, then the transaction will be declined and a status code ‘403’ will be returned to the Callback Message or Status Inquiry API. In these scenarios, it is a business decision on how to properly handle the customer experience, but the transaction will not be authorized by Digital Buy until the address passed matches the customer record.

If the intention is to allow shipping to a different address than the address on file, it is strongly recommended that a strong Know-Your-Customer process is followed.

4.5.2.3.6 Customer Terminated Scenario

If the customer closes the modal without the authorization completing successfully, then a status code ‘100’ will be to the Callback Message or Status Inquiry API. In this scenario, it is recommended that the customer

be returned to tender selection to try again or select a different tender type.

4.5.2.4 Authorization Exception Handling

In some cases, Digital Buy will return a status code of '500' to the Callback Message or Status Inquiry API. This indicates that there was a system error. This error should be logged, and if occurring regularly, escalated for resolution as intervention may be necessary to restore functionality. When these scenarios do occur for the Order Completion Modal, the user experience should be handled in the same way as a customer termination scenario. An error message should be shown, and the user returned to tender type selection in checkout.

4.5.2.5 Authorization Timeout Error Handling

Rarely, a response may not be received for a specific token. The cause for this could be due to network failures or configuration settings, but it should generally be rare. Nevertheless, to ensure a smooth user experience, the web processes handling the individual user checkouts should have a timeout period when waiting for a response.

While the specific timeout period can vary based on business decisions, it is generally recommended that the timeout period to receive and process the first postback message be set to no longer than 30 seconds after the Order Completion Modal closes.

4.6 Combined Modal

The combined modal is used when the customer does not choose to click the "Pay With" button at the cart level, and thus the authentication and Consumer Account Information modal have not yet been called. When the private label or CareCredit card is selected at tender select (instead of with the Rapid Checkout button), this

mode allows one call to Digital Buy to both authenticate the consumer and submit the transaction. It is necessary to have the full order details, including the final amount, promotional code, and customer address to use the Combined Modal mode.

4.6.1 Combined Modal Execution

The Combined Modal is opened by calling the "callCombinedModaldBuyProcess" function (included in the Digital Buy library files) with an argument of a hidden HTML form containing the parameters to be passed to the modal window. It must be called from within a element with a tag id of "dbuymodel3".

Parameter	Type	Data Length	Required	Comments
processInd	Integer	1	**Yes**	Must be passed as "3" to indicate that it is the Combined modal
tokenId	String	29	**Yes**	Correlates to the "clientToken" returned in the authenticate request. This value uniquely identifies the specific checkout
merchantID	String	16	**Yes**	Unique identifier assigned by SYF to identify the specific merchant
clientTransId	String	30 Max	Yes	Merchant generated unique identifier that will be echoed back in the modal responses. meter must be unique, and must be passed as it will be the only way that responses can be isolated in the case of expired or bad tokens.
custFirstName	String	20	No	When present, this parameter will be used to prefill the customer's first name for account lookup and credit application entry. Alphabetic only.
custLastName	String	25	No	When present, this parameter will be used to prefill the customer's first name for account lookup and credit application entry. Alphabetic only.

Parameter	Type	Data Length	Required	Comments
custZipCode	String	5 or 9	Yes	When present, this parameter will be used to prefill the customer's first name for and credit application account lookup entry. Numeric only.
cardNumber	String	16	No	When present, this parameter will be used to prefill the customer's account number for direct account number entry.
iniPurAmt	String	6	No	When present, this parameter will be used to prefill the initial purchase amount for credit application entry.
custAddress1	String	25	Yes	When present, this parameter will be used to prefill the first line of the customer's address for credit application entry.
custAddress2	String	25	Cond.	When present, this parameter will be used to prefill the second line of the customer's address for credit application entry.
phoneNumber	String	10	No	When present, this parameter will be used to prefill the customer's phone number for credit application entry.
emailAddress	String	60	No	When present, this parameter will be used to prefill the customer's email address for credit application entry.
custCity	String	20	YES	When present, this parameter will be used to prefill the customer's city for credit application entry.
custState	String	2	YES	When present, this parameter will be used to prefill the customer's state for credit application entry.
transPromo1	String	3	**Yes**	The three-digit numeric "promo code" that indicates what special financing offer should be applied to "transAmount1". This number should be provided by the Relationship Manager if not known.
transAmount1	BigDecimal	6.2	**Yes**	The amount to be authorized under the promotion passed in "transPromo1". Must include both decimal places and amounts up to 999999.99 are accepted.

Parameter	Type	Data Length	Required	Comments
transPromo2	String	3	Cond.	Optional field that should only be sent if multiple finance offers are in use for the same transaction. The three-digit numeric promotional code to apply to "transAmount2". Must be included in if "transAmount2" is included.
transAmount2	BigDecimal	6.2	Cond.	Optional field that should only be sent if multiple finance offers are in use for the same transaction. The amount to be authorized under the promotion passed in "transPromo1". Must include both decimal places and amounts up to 999999.99 are accepted.
transPromo3	String	3	Cond.	Optional field that should only be sent if multiple finance offers are in use for the same transaction. The three-digit numeric promotional code to apply to "transAmount3". Must be included in if "transAmount2" is included.
transAmount3	BigDecimal	6.2	Cond.	Optional field that should only be sent if multiple finance offers are in use for the same transaction. The amount to be authorized under the promotion passed in "transPromo3". Must include both decimal places and amounts up to 999999.99 are accepted.
defaultPromoCode	String	3	Cond.	Optional field that should only be sent if multiple special financing offers are being applied in the same transaction. This three-digit numeric field indicates the default promotion that should be applied for authorization purposes. The Relationship Manager will provide this value if used.

Combined Modal Window Example

```
<div id="dbuymodel3"></div>

<form name="dbuyform3" id="dbuyform3">
        <input type="hidden" name="processInd" value="3" />
        <input type="hidden" id="TokenId" name="tokenId" />
        <input type="hidden" id="MID" name="merchantID" />
        <input type="hidden" id="CTID" name="clientTransId" />
        <input type="hidden" id="AN" name="cardNumber" />
        <input type="hidden" id="EM" name="expMonth">
        <input type="hidden" id="EY" name="expYear">
```

```
            <input type="hidden" id="iniPurAmtId" name="iniPurAmt">
            <input type="hidden" id="FN" name="custFirstName" />
            <input type="hidden" id="LN" name="custLastName" />
            <input type="hidden" id="zip" name="custZipCode" />
            <input type="hidden" id="custAddress1Id" name="custAddress1">
            <input type="hidden" id="custAddress2Id" name="custAddress2">
            <input type="hidden" id="cityId" name="custCity">
            <input type="hidden" id="stateId" name="custState">
            <input type="hidden" id="phoneNumberId" name="phoneNumber">
            <input type="hidden" id="emailAddressId" name="emailAddress">
            <input type="hidden" id="transPromo1" name="transPromo1" />
            <input type="hidden" id="transAmount1" name="transAmount1" />
            <input type="hidden" id="transPromo2" name="transPromo2" />
            <input type="hidden" id="transAmount2" name="transAmount2" />
            <input type="hidden" id="transPromo3" name="transPromo3" />
            <input type="hidden" id="transAmount3" name="transAmount3" />
            <input type="hidden" id="defaultPromoCode" name="defaultPromoCode" />
<button type="button" id="checkout" class="btn btn-info btn-lg" data-toggle="modal" data-target="#digBuyModal"
            onclick="callCombinedModaldBuyProcess(this.form);">Checkout  </button>
</form>
```

4.6.1 Combined Modal Response Handling

The webpage serving the consumer should detect the closure of the modal window. Once the consumer completes their activity within the modal and it is closed, an immediate call should be made to the Status Inquiry API.

Modal Window Closure Detection Example Code Wrapper

```
$(document).on('hidden.bs.modal','#digBuyModal', function () {

    // findStatus API call to obtain results

});
```

4.6.1.1 Response Format

The result of the Combined modal will be returned to the Status Inquiry API in JSON format with the following elements:

Parameter	Type	Data Length	Comments	Sent In Callback	Sent In API Call / Postback

TokenId	String	29	This parameter correlates to the "clientToken" returned in the authenticate request. This value uniquely identifies the specific checkout	Yes	Yes
StatusCode	Integer	2 or 3	This vaue identifies the result of the attempt to get the consumer's account information in the Combined Modal	Yes	Yes
StatusMessage	String	51 Max	This value is a string representation of the result indicated by StatusCode above	Yes	Yes
FirstName	String	20 Max	When an account is successfully obtained, this parameter contains the cardholder's first name	Yes	Yes
LastName	String	25 Max	When an account is successfully obtained, this parameter contains the cardholder's last name	Yes	Yes
Address1	String	25 Max	When an account is successfully obtained, this parameter contains the cardholder's address (line 1)	Yes	Yes
Address2	String	25 Max	When an account is successfully obtained, this parameter contains the cardholder's address (line 2)	Yes	Yes
City	String	20 Max	When an account is successfully obtained, this parameter contains the cardholder's city	Yes	Yes
State	String	2 Max	When an account is successfully obtained, this parameter contains the cardholder's state	Yes	Yes
zipCode	String	5 or 9 Max	When an account is successfully obtained, this parameter contains	Yes	Yes

			the cardholder's ZIP code		
ClientTransactionID	String	30 Max	This field echoes back the "clientTransId" sent in the Combined Modal request	Yes	Yes
TransactionAmount	String	6.2	This field provides the total amount for the transaction	Yes	Yes
TransactionDate	String	30 Max	The date and timestamp of the transaction – Example: Mon Sep 12 00:45:33 PDT 2016	Yes	Yes
TransactionDescription	String	Max 13	This field is a string representation of whether the transaction was processed as an "AUTHORIZATION" or a "PURCHASE"	Yes	Yes
AuthCode	String	6 Max	This field contains the six-digit numeric authorization code for the transaction. Example: 013798	Yes	Yes
accountNumber	String	16	When the transaction is run in Authorization mode, and is successful, the full unmasked account number will be sent in this field for later use in settlement. *This field is not sent unmasked if tokenization is enabled.*	Yes – Always Masked – See comments	Yes – Full Acct Number in Auth Mode – See Comments
accountToken	String	24	This field returns the account token that can be used later to settle the transaction. It replaces the unmasked account number when used.	No	Yes
PromoCode	String	3 Max	This field echoes back the three-digit numeric code that was applied to "transAmount1", or the default promotional code for transactions	Yes	Yes

			with multiple promotions in use		
FirstName	String	20 Max	The customer's first name	Yes	Yes
LastName	String	25 Max	The customer's last name	Yes	Yes
PostbackId	String		This field contains the postbackid that is sent in the initial authentication response. It should be compared to the value stored security on the server to confirm that the response is from SYF and not a third party.	No	Yes

Status Code	Status Message
400	Required input not provided/Input validation error (Token not found in DB)
401	Transaction Failed (Token Expired Scenario)
100	Customer Terminated
04, also possible (06, 99)	eApply_Transaction_Pending
07, also possible (08, 12, 13, 14, 15, 16, 17, 18, 81, 82, 90)	eApply_Transaction_Declined
09 or NO Code Sent	eApply_CALL_NOW
03, 10, 11	eApply_PROCESS_ERROR
500	System Exception
000	Authorization Approved/ Purchase Success
001	Authorization Declined/Purchase Declined
402	Promo code validation fail
403	Address verification check Fail

4.6.1.1 Account Authentication Success Scenario

4.5.2.3 Apply Scenario

When a customer does not already have an account, they may choose to apply for one from the Account Information modal. In these cases, a credit application will open in a separate tab, and the modal will remain open on the merchant website with a message stating “Credit application in process, please complete the application in the other browser window” and an option to close the modal should they decide to not apply.

4.5.2.4 .1 Apply Scenario – Approved

Once that customer completes the application process and is approved for a new account, the transaction will continue.

4.5.2.4.2 Apply Scenario – Declined

For scenarios where the customer is declined for a new account, the modal will close and the data returned via Callback Message or findStatus API call will contain a status code of ‘**07’**. While rare, it is also possible to receive 08, 12, 13, 14, 15, 16, 17, 18, 81, 82, or 90 in the application declined scenario. In these cases, it should be treated as a failure to obtain an account number, and the user returned to the tender selection portion of the checkout process.

4.5.2.4.3 Apply Scenario - Pending

On rare occasions, there will be a scenario where Synchrony will be unable to reach an immediate decision on the customer's credit application (an example of this would be if the customer has a hold on their credit bureau). When this occurs, the credit application will be placed into a 'pending' state and the data returned via Callback Message or findStatus API call will contain a status code of '**04'**. While rare, it is also possible to receive 06 or 99 in the application pending scenario. When these scenarios occur, they must be treated as a failure to obtain an account number, and the user returned to the tender selection portion of the checkout process.

4.5.2.4.4 Apply Scenario – Exception Responses

On very rare occasions, there could be a scenario where an internal systems issue might result in a situation where an application was not processed for the customer. When this occurs, the credit application will be placed into a 'pending' state and the data returned via Callback Message or findStatus API call will contain a status code of '03', '09' '10' or '11'(*see table for status message). Very rarely, an empty '' status code with the Status Message of 'eApply_CALL_NOW' can be sent. This should only occur during an outage situation. When these scenarios occur, they must be treated as a failure to obtain an account number, and the user returned to the tender selection portion of the checkout process.

4.6.1.2 Account Authentication Failure Scenario

If Digital Buy is not able to successfully obtain an account number for the customer with the then it will return one of several responses that can all be treated as a failure scenario. These cases include '400' (input validation error

or token not found in database), ‘401’ (token expired), and ‘100’ (user terminated modal without providing an account number). For each of these cases, the user should be given an error message and returned to tender type selection.

4.6.1.3 Exception Handling

In some cases, Digital Buy will post a status code of ‘500’ via Callback Message or Status Inquiry API. This indicates that there was a system error. This error should be logged, and if occurring regularly, escalated for resolution as intervention may be necessary to restore functionality. When these scenarios do occur, the user experience should be handled in the same way as an account authentication failure. An error message should be shown, and the user returned to tender type selection in checkout.

4.6.1.4 Timeout Error Handling

Rarely, a response may not be received by the Status Inquiry API for a specific token. The cause for this could be due to network failures or configuration settings, but it should generally be an edge case. Nevertheless, to ensure a smooth user experience, the web processes handling the individual user checkouts should make multiple attempts to call the API and have a timeout period when waiting for a response.

While the specific timeout period can vary based on business decisions, it is generally recommended that the timeout period to receive and process response from the findStatus API call be set to no longer than 30 seconds after the modal closes.

4.6.1.5 Authorization Successful Scenario

In cases where the Combined Modal successfully obtains an authorization, the Digital Buy platform will return a status code ‘000’ to the Status Inquiry

API. In these cases, the customer's purchase can be considered complete for the amount sent. When appropriate, the cart should be cleared and the customer allowed to continue with a fresh order.

If Digital Buy is running in Purchase mode, settlement will occur automatically overnight. If it is running in Authorization mode, then the full account number must be retrieved at the end of the transaction when the findStatus API is called. It is required for settlement later (through a different channel) by the merchant.

4.6.1.6 Authorization Failure Scenario

The Order Completion step of the Combined Modal purchase process can fail for several different reasons. Each of these may potentially need to be handled differently depending on the specific checkout environment.

4.6.1.6.1 Authorization Declined

An Authorization Declined response (status code '001') occurs when the customer's account is not approved for the transaction as sent. In this scenario, it is necessary to return the user to tender select so they can choose an alternate tender type in order to complete the order.

4.6.1.6.2 Promotional Code Validation Error Scenario

When the three-digit numerical code that controls what special financing should be applied to the order is unable to be validated, a status code '402' will be returned via Callback Message or Status Inquiry API. This error indicates that either a promotional code has been configured incorrectly, or an existing promotional code has expired. These errors should be escalated to your primary point-of-contact at Synchrony as they typically require manual intervention to resolve.

In this scenario, the customer should be returned to tender selection.

4.6.1.6.3 Address Verification Failure Scenario

If the customer address passed to the Order Completion does not match the address on the SYF systems, then the transaction will be declined and a status code '403' will be returned via Callback Message or Status Inquiry API and/or listener URL. In these scenarios, it is a business decision on how to properly handle the customer experience, but the transaction will not be authorized by Digital Buy until the address passed matches the customer record.

If the intention is to allow shipping to a different address than the address on file, it is strongly recommended that a strong Know-Your-Customer process is followed.

4.7 Callback Status Messages

When a modal window is closed, Digital Buy can be configured to pass non-sensitive status data to the merchant site as a callback message.

4.7.1.1 Response Format

The result of the modal outcome will be passed via callback message in JSON format. A list of parameters and data requirements can be found in the details of each modal mode section.

Callback Status Message Example – Java

```
@RequestMapping("/parseDbuyJsonCallBack")
```

```
	public void parseDbuyJsonCallBack(HttpServletRequest httpServletRequest,
@RequestParam("callbackMessage") String callbackMessage) {
		System.out.println("-----------------ParseDbuyJsonCallBack---------------
----");
		if (callbackMessage != null) {
			try {
				JSONObject jsonObj = new JSONObject(callbackMessage);
				String statusCode = jsonObj.getString("StatusCode");
				String statusMessage = jsonObj.getString("StatusMessage");
				String tokenId = jsonObj.getString("TokenId");

				System.out.println(statusCode);
				System.out.println(statusMessage);
				System.out.println(tokenId);

			} catch (Exception e) {
				e.printStackTrace();
			}
		}
	}
```

4.8 Status Inquiry API

Whenever a modal window is detected as closed, a server-side call to the Status Inquiry API should be initiated to obtain the results of the execution.

API Endpoints

Test Region: https://usvcs.syf.com/v1.0/status/inquiry

Production Region: https://svcs.syf.com/v1.0/status/inquiry

The body of the secure API Call should be in JSON format and contain the following name-value pairs:

- "**merchantNumber**" and the corresponding Merchant ID (MID) provided by SYF
- "**password**" and the corresponding password provided by SYF
- "**userToken**" and the value for the clientToken for the transaction you want the data returned for.

The password parameter should never be exposed to the public, and any code containing the cleartext values must be properly secured.

Upon successful receipt of valid credentials and userToken, the findStatusAPI will respond to the call with data generated from the last modal event in JSON format.

4.8.1 findStatus API Response Parameters

The following parameters returned in the API call are specific to the identificatio and success of the call itself.

transactionId: A GUID that is sent with every call to the API, it is unique to each call and never duplicated, even if the data does not change.

responseCode: "000" indicates a successful call to the API. Any other code indicates a failure.

"responseDesc": "SUCCESS" indicates a successful call, anything else indicates a failure.

“errors”: This field will be present when additional errors are generated with the details of those errors.

The transactional parameters returned are outlined in the Modal Response sections.

Sample code for the Status Inquiry API execution is given below in both Java and PHP.

Status Inquiry API Example – Java

```
public static void client() {

        String data = null;
        String output = null;
        List<InquiryPostBackRequestVO> inquiryPostBackRequestVOList = null;
        ObjectMapper mapper = null;
        Client client = null;
        WebResource webResource = null;
        ClientResponse response = null;
               // Instantiate InquiryPostBackRequestVO
        InquiryPostBackRequestVO inquiryPostBackRequestVO = new
InquiryPostBackRequestVO();
        inquiryPostBackRequestVO.setMerchantNumber("MerchanNumber");
        inquiryPostBackRequestVO.setPassword("Password");
        inquiryPostBackRequestVO.setUserToken("TokenId");


        inquiryPostBackRequestVOList = new ArrayList<InquiryPostBackRequestVO>();
        inquiryPostBackRequestVOList.add(inquiryPostBackRequestVO);

        try {
               mapper = new ObjectMapper();
               data = mapper.writeValueAsString(inquiryPostBackRequestVOList);
               data=data.substring(1,data.length()-1);
               System.out.println(data);

        } catch (JsonGenerationException e) {
               e.printStackTrace();
        } catch (JsonMappingException e) {
               e.printStackTrace();
        } catch (IOException e) {
               e.printStackTrace();
        }
```

```
                //Create client and set resource with the Inquiry service URL
        client = Client.create();

                //Test\QA\Dev Endpoint
                webResource =
client.resource("https://usvcs.syf.com/v1.0/status/inquiry");

                //Production Endpoint
                //webResource =
client.resource("https://svcs.syf.com/v1.0/status/inquiry");

                //Response from webservice
        response = webResource.type("application/json").post(ClientResponse.class,
data);

                //Actual output or response data
        output = response.getEntity(String.class);
        System.out.println("Output from Server : \n");
        System.out.println(output);
 }

 ================================================================================
// Bean to be used in client code: -

InquiryPostBackRequestVO bean should created to use in Java client code with below
mentioned fields: -
    private String merchantNumber;
    private String password;
    private String userToken;

================================================================================
```

Status Inquiry Example – Java

```
<?php

// Set MerchantNumber, Password, TokenId
$merchantNumber = '<MerchantNumber>';
$password = 'Password';
$userToken = 'TokenId';

// data needs to be POSTed to the Play url as JSON.
$data = array("merchantNumber" => "$merchantNumber", "password" => "$password",
"userToken" => "$userToken");
$data_string = json_encode($data);

//Initializes a new session and return a cURL handle for use

//Test\QA\Dev Endpoint
$ch = curl_init(' https://uatcdws.app.syfbank.com/StatusInquiry/v1/findStatus');

//Production Endpoint
// $ch = curl_init('https://svcs.syf.com/v1.0/status/inquiry');

//cURL session handle
```

```
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string))
);
curl_setopt($ch, CURLOPT_TIMEOUT, 3);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 3);

//execute post
$result = curl_exec($ch);

//close connection
curl_close($ch);

//Print result
echo $result;

?>
```

4.9 Callback Redirection

Digital Buy contains the option to direct the user to a successfulPathURL or failurePathURL supplied by the merchant each time a modal closes.

This is accomplished by passing the URLs as input to the redirectToNextPage function when calling the modal.

- **Consumer Account Information Modal:** Function input for failure path only.
- **Order Completion Modal:** Function inputs to redirect to successful path and failure path.
- **Combinded Modal:** Function inputs to redirect to successful path and failure path.

Digital Buy Callback Redirection

Example:

Customer Information Modal (Modal 1)

```
<div class="row">
        <div class="col-xs-12 text-center">
        <button type="button" id="checkout" class="btn btn-info btn-lg"data-
toggle="modal" data-target="#digBuyModal"
        onclick="calldBuyProcess1(this.form,redirectToNextPage('auth success
url'));">Checkout</button>
</div>

Order Completion Modal (Modal 2)
<div class="col-xs-10" style="padding-top: 10px">
        <div class="col-xs-6" style="text-align: right;">
        <button type="button" id="finalSubmit" class="btn btn-info"
        data-toggle="modal" data-target="#digBuyModal"
        onclick="calldBuyProcess2(this.form,redirectToNextPage('transaction success
        url','transaction failure url'))">Continue</button>
</div>
```

5 Troubleshooting

This section lists common pitfalls that have been reported when integrating to Digital Buy. Should you encounter a scenario that is not listed here, it is asked that you provide that feedback to Synchrony during the post-implementation review so that this documentation can be improved for future integrations.

5.1 Authentication Attempts Failing with '401-Unauthorized'

This error usually indicates that the domain from which authentication is being attempted has not been whitelisted by the SYF Digital Buy platform. When this error is received, it is recommended to confirm that the domain being used has been added to the SYF whitelist. This can be confirmed by contacting your primary point-of-contact at Synchrony, who will work with the SYF IT POS Integration Manager.

This response can also occur if the Digital Buy credentials ("merchantID" and "password") are sent as standard HTTPS authentication. These credentials should be sent in the body of the post as their own individual parameters per the Authentication section of this guide.

5.2 Not Receiving Responses ("Postbacks")

For the Digital Buy platform to respond to modal activity, it is necessary to configure a listener URL that is capable of receiving secure posts. Scenarios where modal activity occurs but no post is received are usually the result of one of the following:

- The listener URL has not been properly configured for the Digital Buy credentials. If this is the case, this can be resolved by providing the correct listener URL to the IT POS integration manager, who will have it configured appropriately.

- The listener URL has been configured to require authentication to deliver a post. This can be corrected by opening the listener URL to accept posts without authentication. As the listener will be open to all posts, it should be stressed that the 'postbackid' (provided during the initial Digital Buy Authentication request and stored securely on the server) be verified before shipping product to make sure that the post response is legitimate.

6 Glossary of Terms

This section lists common terms that are used throughout this document.

Authorization Mode: This term refers to how Digital Buy processes transactions. When in 'Authorization Mode', successful transactions are not automatically settled. Instead, a hold on the customer's open-to-buy will be enacted for a set number of days (industry specific) and settlement/completion of the transaction must occur through another channel.

Bootstrap: Bootstrap is a popular open-source development library. Digital Buy leverages functionality from this library.

CSS: CSS stands for Cascading Style Sheet. It is a mechanism for defining how web elements look and are formatted. Digital Buy requires inclusion of the Digital Buy CSS to function properly.

Deferred Interest Financing: Deferred Interest financing is a type of special financing where interest on the person is accrued from the date of purchase, but not charged until the promotional period ends. If the purchase is paid off prior to the end of the promotional period, then the interest is not assessed.

Domain: In this context, domain refers to the hostname of a web page accessible to the internet. As an example, synchronyfinancial.com is a domain.

HTTP(S) Post: A post is a standard method within the Hypertext Transfer Protocol. It is used to deliver data to a web page address.

Integration Region: The integration region is a fully isolated environment that can be used for testing and integration work. Data is not shared between the integration region and the production region.

JavaScript: JavaScript is a popular programming language used in web page development (as well as many other applications).

JSON: JSON stands for JavaScript Object Notation and is a standardized data format for sending information using attribute-value pairs in a human readable format.

Listener: Listener is the term used for a URL that is configured to accept incoming HTTP(S) posts. When those posts are received, a trigger routine is executed to process the data.

Modal: A modal is a dialog window that displays on top of the current page, but within the same window (from the perspective of the user).

No Interest Financing: No interest financing is a type of special financing where the consumer is never charged interest on the purchase, but could be liable for other fees (if applicable) based on the credit terms.

PostbackId: The postbackId is a parameter returned in the authentication response message. This value should be stored securely on the server, and never revealed to the consumer, as it is the mechanism used to validate that the final authorization response is truly from Synchrony, and not from an attacker trying to spoof a result.

Production Region: The production region is the environment where live transactions and credit applications are processed.

Promotion: A promotion refers to a special financing offer, where the terms of the purchase are different than simple revolving. Typically, promotions are offered as either deferred interest, reduced APR, or no interest.

Promotion Code: The promotion code is the three digit numeric field that indicates which special financing offer (promotion) to apply to the purchase.

Promotional Strategy: The promotional strategy term refers to the way in which special financing offers are handled within the shopping cart. As an example, the promotional strategy for a shopping cart might be to give all purchases a 6 month deferred interest financing offer.

Purchase Mode: When Digital Buy is run in “Purchase Mode”, any successful transactions will be automatically settled that night.

Rapid Checkout: Rapid checkout refers to the ability to click a ‘pay with’ button at the shopping cart level. This allows the customer to initiate finalization of the transaction with a specific tender type early in the process rather than as one of the final steps.

Reduced Interest Financing: Reduced Interest financing is a type of special financing where the purchase is charged a reduced interest rate during the promotional period. Fixed payments are required during the period.

Settlement: Settlement refers to the process where a customer is billed for the transaction, and the merchant is paid.

Tender Selection: Tender selection refers to the portion of the checkout where the customer chooses the method with which they wish to pay. As an example, they may have the option of choose between

the private label or CareCredit card, a bank card (VISA, Mastercard, etc.), or a gift card.

White List: In this context, “white list” refers to the list of domains with which Synchrony will allow processing to the Digital Buy platform.

7 License Terms

SYNCHRONY BANK LICENSE TERMS

DIGITAL BUY SOFTWARE

These license terms ("Agreement") are a legally binding contract between Synchrony Bank ("Issuer") and the retailer/dealer/merchant associated with a Card account ("Retailer", "you" or "your") who is accessing the Digital Buy Software. Please read carefully the terms of this Agreement as you agree to them by assessing and downloading the Digital Buy Software.

This Agreement supplements the Web Site Usage Agreement found on the Synchrony Bank website. Capitalized terms used and not otherwise defined in this Agreement will have the meanings ascribed to them in the Web Site Usage Agreement. In the event of a direct conflict between language included in this Agreement and language included in the Web Site Usage Agreement, the provisions of this Agreement will control solely with respect to application to the Digital Buy Software, Digital Buy Documentation and Digital Buy Functionality.

If you comply with the terms of this Agreement, you have the rights detailed below.

1. DEFINITIONS

a. "Claim(s)" means all third-party claims, actions, demands, proceedings, damages, costs and liabilities of any kind.

b. "Digital Buy Documentation" means the end user documentation, provided or made available by Issuer to Retailer along with the Digital Buy Software, including, where applicable, any updates or future versions of the Digital Buy Documentation that Issuer provides or makes available to you.

c. "Digital Buy Functionality" means the functionality enabled by the Digital Buy Software that allows Retailer's customers to pay Issuer (and receive promotional financing from Issuer, if available) from the Retailer Site.

d. "Digital Buy Software" means the software provided to you under this Agreement, and specifically including application program interfaces ("APIs"), including, where applicable, any upgrades, updates, or future versions of the Digital Buy Software that Issuer provides to you.

e. "Retailer Site" means your customer-facing web site that provides the Digital Buy Functionality.

2. **LICENSE RIGHTS.** Subject to all of the terms of this Agreement, and to Retailer's continuing compliance with this Agreement, Issuer grants Retailer a limited, non-exclusive, personal, non-transferable, non-assignable, and terminable license to:
   a. reproduce a reasonable number of copies of the Digital Buy Software for your internal use in exercising the rights granted to you in this Section 2;
   b. use the Digital Buy Software internally only and solely for the development and testing of the Digital Buy Functionality;
   c. reproduce the object code form of the Digital Buy Software as part of the Digital Buy Functionality on the Retailer Site; and
   d. reproduce and use the Digital Buy Documentation in connection with the use of the Digital Buy Software solely for Retailer's internal business purposes.
3. **RESTRICTIONS.** The following terms apply to all license grants related to the Digital Buy Software.
   a. **Compliance with Digital Buy Documentation.** Retailer will use the Digital Buy Software and implement the Digital Buy Functionality in accordance with the terms of the Digital Buy Documentation.
   b. **Testing.** Retailer will complete the testing required by the Digital Buy Documentation and the Site before making the Digital Buy Functionality available to your customers on the Retailer Site.
   c. **Limited Use.** Issuer does not grant you the right to use the Digital Buy Software in connection with the development of other software, services or products for any purpose other than for developing and using the Digital Buy Functionality.
   d. **Licensed Not Sold.** All use and distribution of the Digital Buy Software is by license only. You must comply with any technical limitations in the Digital Buy Software that only allow you to use it in certain ways.
   e. **Copying.** You may reproduce only complete copies of the Digital Buy Software, including all "read me" files, copyright notices, and other legal notices and terms that Issuer has included in the Digital Buy Software.
   f. **No Distribution.** You may not distribute or transfer the Digital Buy Software or any copy you make of the Digital Buy Software to any other party except your company's subsidiaries.
   g. **IP Notices.** You will not remove or obscure any copyright, trademark, or patent notices that appear on the Digital Buy Software or Digital Buy Documentation as delivered to you.
   h. **Export Restrictions.** The Digital Buy Software is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Digital Buy Software.

i. **Additional Restrictions.** You may not:

   i. **work around any technical limitations in the Digital Buy Software;**

   ii. **modify, reverse engineer, decompile or disassemble the Digital Buy Software, except and only to the extent that applicable law permits, despite this limitation;**

   iii. **publish the Digital Buy Software for others to copy, except for vendors you may engage to help you test, deploy or develop the Digital Buy Functionality;**

   iv. **sell, rent, lease or lend copies of the Digital Buy Software;**

   v. **use the Digital Buy Software for commercial software hosting services;**

   vi. **use the Digital Buy Software or any portion thereof to implement any products or services to operate in connection with the Digital Buy Software for any other purpose than granted herein; or**

   vii. **disclose the results of any benchmark tests of the Digital Buy Software to any third party without Issuer's prior written approval.**

4. **CHANGES AND SUPPLEMENTS.** You understand and agree that Issuer may amend, modify, change, supplement, and cease distribution or production of the Digital Buy Software at any time. You understand that you are not entitled to receive any supplements, upgrades, updates, or future versions of the Digital Buy Software under this Agreement.

5. **PROMOTIONAL TERMS.** Retailer will provide to Issuer in a timely and complete manner accurate promotional codes for the promotions Retailer desires to offer to customers in connection with the Digital Buy Functionality to permit Issuer to determine which of Retailer's products qualify for promotions and the terms of such promotions.

6. **FEES.** No payments will be owed by either party to the other for the license and rights granted under this Agreement. Each party will be responsible for any applicable taxes, fees, or expenses incurred by such party pursuant to this Agreement.

7. **OWNERSHIP.** Except for the licenses expressly granted under this Agreement, Issuer retains all rights, title and interest in and to the Digital Buy Software, Digital Buy Documentation and all intellectual property rights therein.

8. **CONFIDENTIAL INFORMATION.** The Digital Buy Software and Digital Buy Documentation are confidential and proprietary to Issuer and its suppliers.

   a. **Use.** You may not disclose confidential information to third parties. You may disclose confidential information only to your employees and independent contractors who need to know the information. You must have written agreements with them that protect the confidential information with terms at least as stringent as contained in this Agreement.

   b. **Survival.** Your duty to protect confidential information survives this Agreement.

   c. **Exclusions.** You may disclose confidential information in response to a judicial or governmental order. You must first give written notice to Issuer to allow it to seek a

protective order or otherwise protect the information. Confidential information does not include information that

- becomes publicly known through no wrongful act;
- you received from a third party who did not breach confidentiality obligations to Issuer or its suppliers; or
- you developed independently.

9. **FEEDBACK.** You may provide comments, suggestions, or proposed improvements to Issuer about the Digital Buy Software or the Digital Buy Functionality (collectively, "Feedback"). You are not required to provide any Feedback to Issuer. If you provide Feedback to Issuer, you give to Issuer, without charge, the right to use, share and commercialize the Feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of an Issuer product or service that includes the Feedback. You will not give Feedback that is subject to a license that requires Issuer to license its software or documentation to third parties because Issuer includes your Feedback in them. These rights survive this Agreement.
10. **CHANGES TO AGREEMENT.** If Issuer changes this Agreement, you will be given notice before the change is in force. If you do not agree to these changes, then you must stop using the Digital Buy Software before the changes are in force. If you do not stop using the Digital Buy Software, then your use of the Digital Buy Software will continue under the changed contract. Issuer may give notices to you by electronic mail to any e-mail address provided by you to Issuer and/or by posting such notices on the Site.
11. **TERMINATION.** Issuer may, at any time, terminate this Agreement with you, if (a) you have breached any provision of this Agreement; or (b) Issuer is required to do so by law. Further, Issuer reserves the right to discontinue offering the Digital Buy Software or certain parts of the Digital Buy Software or to modify the Digital Buy Software at any time in its sole discretion, and it may suspend access to any APIs that are part of the Digital Buy Software to you or any end user at any time. Upon the termination of this Agreement, you will discontinue all use of the Digital Buy Software, Digital Buy Documentation and Digital Buy Functionality. This Section 11 and Sections 1, 3, 4, 7, 8, 9, 12, 13, 14 and 15 will survive termination of this Agreement or any discontinuance of the offering of the Digital Buy Software, along with any other provisions that would reasonably be deemed to survive such events.
12. **DISCLAIMER OF WARRANTIES. The Digital Buy Software is licensed "as is." You bear the risk of using it. Issuer gives no express warranties, guarantees or conditions. To the extent permitted under your local laws, Issuer excludes the implied warranties of merchantability, fitness for a particular purpose and non-infringement.**
13. **INDEMNIFICATION.** You will defend, indemnify, and hold Issuer, its affiliates, and their respective successors, directors, officers, employees, and agents harmless from and against all Claims to the extent that such Claims arise out of or relate to (i) any infringement,

misuse, or misappropriation of any third-party copyright, trademark, trade secret, trade dress, patent or other intellectual property rights (collectively, "Proprietary Rights") by the Retailer Site, (ii) any non-compliance by you of this Agreement, or (iii) your negligence or willful misconduct. You agree not to stipulate, admit, or acknowledge any fault or liability on the part of Issuer without Issuer's prior written consent, and you agree not to settle any Claim or publicize any settlement without Issuer's prior written consent.

14. **LIMITATION ON AND EXCLUSION OF DAMAGES. You can recover from Issuer and its suppliers only direct damages up to US$250.00. You cannot recover any other damages, including consequential, lost profits, special, indirect or incidental damages.**

    This limitation applies to

    - anything related to the Digital Buy Software, the Digital Buy Documentation, and the Digital Buy Functionality; and
    - claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

    It also applies even if Issuer knew or should have known about the possibility of the damages.

    Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you. They also may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

15. **MISCELLANEOUS**

    a. **Applicable Law.** This Agreement will be governed by the laws of the State of New York and the federal laws of the United States of America, without giving effect to their conflict of laws provisions. For all disputes arising from or related to the Agreement, you agree to submit to the personal and exclusive jurisdiction of the state and federal courts located in New York.

    b. **Entire Agreement.** This Agreement is the entire agreement for the Digital Buy Software, Digital Buy Documentation and Digital Buy Functionality.

    c. **Relationship of the Parties.** The parties agree that the Agreement will not be construed as creating a partnership, joint venture or agency relationship or as granting a franchise.

    d. **No Waiver.** No waiver of any breach of any part of this Agreement will be a waiver of any other breach. Any waiver must be in writing and signed by an authorized representative of the waiving party.

    e. **No Assignment.** You may not assign or transfer your rights or obligations under this Agreement to a third party without Issuer's prior written approval.

f. Severability. If a court of competent jurisdiction finds any term of this Agreement illegal, invalid or unenforceable, the remaining terms will remain in full force and effect.

g. English Language. This Agreement is written only in the English language, which language will be controlling in all respects. Versions of the Agreement in any other language are only for accommodation and are not binding on the parties.