**CyberSource SiteGenesis Controllers LINK Cartridge 3DS2.X - Developer Guide**

***Upgrading from Version 17.2.X to Version******19.3.0***

A picture containing clock, drawing

Description automatically generated

­

Table of Contents

[1. CyberSource Cartridge Overview 2](#_Toc60141260)

[2. Payer Authentication 2](#_Toc60141261)

[Payer Authentication Service 6](#_Toc60141262)

[3. Change Logs 8](#_Toc60141263)

[4. Support French Processor 11](#_Toc60141264)

# CyberSource Cartridge Overview

The CyberSource package contains four cartridges. A core cartridge (**int\_cybersource**) that contains core API integrations, including the building and handling of API requests, and parsing responses into objects usable by the storefront. The two legacy architecture cartridges (**int\_cybersource\_pipelines, int\_cybersource\_controllers**) each contain sets of wrappers that connect the core code to their respective SFCC platforms. Version 18.1 and higher of the CyberSource cartridge package adds a fourth cartridge (**int\_cybersource\_sfra/ LINK\_cybersource**), which combines a modified version of the core code that exists in the int\_cybersource, along with the necessary hooks, extensions, and wrappers to connect this code to the SFRA storefront.

# Payer Authentication

The CyberSource LINK Cartridge 19.3.0 provides support for the latest 3D Secure 2.x specification. Upgrading your Payer Authentication integration from the legacy 3D Secure 1.0 specification to 2.x will allow for better authentication rates from issuers and meet PSD2 regulatory requirements for Strong Customer Authentication (SCA) in Europe.

The following is a high-level architectural diagram of how the Payer Authentication integration works:

Chart, waterfall chart

Description automatically generated

*Implementation of 3DS2.x*

#### <Reference Generic section on CyberSource LINK Cartridge Version 19.3.0 integration guide document >

#### Template – summary.isml

Below changes are generic for Secure Accpetance/Klarna\_credit/Device fingerprint

1. Set summary page tag for Secure Acceptance Iframe

|  |
| --- |
| <iscontent type=*"text/html"* charset=*"UTF-8"* compact=*"true"*/>  <isset name=*"summarypage"* value=*"${true}"* scope=*"page"*/>  <isdecorate template=*"checkout/pt\_checkout"*/> |

1. Add below code above <isreportcheckout checkoutstep="${5}" checkoutname="${'OrderSummary'}"/>

|  |
| --- |
| <isscript>  var CybersourceConstants = require('int\_cybersource/cartridge/scripts/utils/CybersourceConstants');  </isscript>  <isset name=*"klarnarequired"* value=*"${false}"* scope=*"page"*/>  <isif condition=*"${!empty(pdict.Basket)}"*>  <isset name=*"LineCntr"* value=*"${pdict.Basket}"* scope=*"page"*/>  <iselseif condition=*"${!empty(pdict.Order)}"*>  <isset name=*"LineCntr"* value=*"${pdict.Order}"* scope=*"page"*/>  </isif>  <isset name=*"summaryaction"* value=*"${URLUtils.https('COSummary-Submit')}"* scope=*"page"* />  <script src=*"${URLUtils.staticURL('/lib/jquery/jquery-1.11.1.min.js')}"* type=*"text/javascript"*></script>  <isset name=*"paymentMethod"* value=*"${null}"* scope=*"page"*/>  <isset name=*"isIFrame"* value=*"${false}"* scope=*"page"* />  <isif condition=*"${!empty(LineCntr.getPaymentInstruments())}"*>  <isloop items=*"${LineCntr.getPaymentInstruments()}"* var=*"paymentInstr"* status=*"loopstate"*>  <isset name=*"paymentMethod"* value=*"${dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod).ID}"* scope=*"page"*/>  <isif condition=*"${dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod).ID==CybersourceConstants.METHOD\_SA\_IFRAME}"*>  <isset name=*"summaryaction"* value=*"${URLUtils.https('COSummary-SubmitOrder')}"* scope=*"page"* />  <isset name=*"isIFrame"* value=*"${true}"* scope=*"page"* />  <iselseif condition=*"${CybersourceConstants.KLARNA\_PAYMENT\_METHOD.equals(dw.order.PaymentMgr.getPaymentMethod(paymentInstr.paymentMethod).ID)}"* >  <isset name=*"klarnarequired"* value=*"${true}"* scope=*"page"*/>  </isif>  </isloop>  </isif>  <isif condition=*"${!empty(LineCntr)}"*>  <isreportcheckout checkoutstep=*"${5}"* checkoutname=*"${'OrderSummary'}"*/> |

1. Replace pdict.Basket with LineCntr at below places

|  |
| --- |
| <isif condition=*"${!pdict.CurrentForms.multishipping.entered.value}"*>  <ischeckoutprogressindicator step=*"3"* multishipping=*"false"* rendershipping=*"${LineCntr.productLineItems.size() == 0 ? 'false' : 'true'}"*/>  <iselse/>  <ischeckoutprogressindicator step=*"4"* multishipping=*"true"* rendershipping=*"${LineCntr.productLineItems.size() == 0 ? 'false' : 'true'}"*/>  </isif> |

1. Add condition for secure acceptance error by replacing place order error with below code

|  |
| --- |
| <div id=*"errordiv"*>  <isif condition=*"${pdict.CurrentHttpParameterMap.SecureAcceptanceError != null && !empty(pdict.CurrentHttpParameterMap.SecureAcceptanceError.stringValue)}"*>  <div class=*"error-form"*>${Resource.msg('confirm.error.technical','checkout',null)}</div>  <iselseif condition=*"${pdict.PlaceOrderError != null}"*>  <div class=*"error-form"*>${Resource.msg(pdict.PlaceOrderError.code,'checkout',null)}</div>  </isif>  </div>  </isif> |

1. Replace pdict.Basket with LineCntr at below places

|  |
| --- |
| <iscomment>render each shipment</iscomment>  <isset name=*"shipmentCount"* value=*"${0}"* scope=*"page"*/>  <isloop items=*"${LineCntr.shipments}"* var=*"shipment"* status=*"shipmentloopstate"*>  <isif condition=*"${shipment.productLineItems.size() > 0 || shipment.giftCertificateLineItems.size() > 0}"*>  <isset name=*"shipmentCount"* value=*"${shipmentCount+1}"* scope=*"page"*/>  <isif condition="${LineCntr.shipments.size() > 1}">  . . . . .  . . . <existing code>. . .  . . . .  <iscomment>RENDER COUPON/ORDER DISCOUNTS</iscomment>  <isloop items="${LineCntr.couponLineItems}" var="couponLineItem" status="cliloopstate">  . . . .  . . <existing code>. .  . . . .  <td class=*"item-total"*>  <isif condition=*"${couponLineItem.applied}"*>  <span class=*"coupon-applied"*>${Resource.msg('summary.applied','checkout',null)}</span>  <iselse/>  <span class=*"coupon-not-applied"*>${Resource.msg('summary.notapplied','checkout',null)}</span>  </isif>  </td>  </tr>  </isif>  </isloop>  <isloop items="${LineCntr.priceAdjustments}" var="priceAdjustment" status="cliloopstate"> |

1. Update with below section for Klarna/Secure acceptance Iframe and device fingerprint and cardinal script related changes

|  |
| --- |
| <div class=*"order-summary-footer"*>  <div class=*"place-order-totals"*>  <isordertotals p\_lineitemctnr=*"${LineCntr}"* p\_showshipmentinfo=*"${false}"* p\_shipmenteditable=*"${false}"* p\_totallabel=*"${Resource.msg('summary.ordertotal','checkout',null)}"*/>  </div>  <isif condition=*"${!empty(klarnarequired) && klarnarequired}"* >  <div id=*"klarna\_container"*></div>  <div id=*"auth\_button"*></div>  <input type=*"hidden"* id=*"processorToken"* name=*"processorToken"* value=*${session.privacy.processorToken}*/>  </isif>  <isif condition=*"${!empty(pdict.Basket)}"*>  <form action=*"${summaryaction}"* method=*"post"* class=*"submit-order"* name=*"submitOrder"*>  <fieldset>  <div class=*"form-row"*>  <a class=*"back-to-cart <isif condition="*${!empty(klarnarequired) && klarnarequired}"> hide</isif>" href="${URLUtils.url('Cart-Show')}">  <isprint value=*"${Resource.msg('summary.editcart','checkout',null)}"* encoding=*"off"* />  </a>  <isif condition=*"${!empty(klarnarequired) && klarnarequired}"* >  <input type=*"hidden"* id=*"klarnaAuthToken"* name=*"klarnaAuthToken"*/>  </isif>  <button class=*"button-fancy-large <isif condition="*${!empty(klarnarequired) && klarnarequired}"> hide</isif>" type="submit" name="submit" value="${Resource.msg('global.submitorder','locale',null)}">  ${Resource.msg('global.submitorder','locale',null)}  </button>  </div>  <input type=*"hidden"* name=*"${dw.web.CSRFProtection.getTokenName()}"* value=*"${dw.web.CSRFProtection.generateToken()}"*/>  <input type=*"hidden"* id=*"DFReferenceId"* name=*"DFReferenceId"* />  </fieldset>  </form>  </isif>  </div>  <isif condition=*"${dw.system.Site.getCurrent().getCustomPreferenceValue('CsDeviceFingerprintEnabled')}"*>  <isinclude url=*"${URLUtils.url('CYBCredit-IncludeDigitalFingerprint')}"*/>  </isif>  <isif condition=*"${isIFrame}"*>  <isinclude template=*"secureacceptance/secureAcceptanceIframeSummmary"*/>  </isif>  <isif condition=*"${pdict.iscardinal }"*>  <isinclude template=*"cardinal/songbird"*/>  </isif>  </isdecorate> |

### Payer Authentication Service

#### Controller - COSummary.js

##### Update submit Function

Update function to handle Payer auth redirection

|  |
| --- |
| **function** submit() {  // Calls the COPlaceOrder controller that does the place order action and any payment authorization.  // COPlaceOrder returns a JSON object with an order\_created key and a boolean value if the order was created successfully.  // If the order creation failed, it returns a JSON object with an error key and a boolean value.  **var** cart = Cart.get();  **var** DFReferenceID = request.httpParameterMap.DFReferenceId.stringValue;  session.privacy.DFReferenceID = DFReferenceID;  **var** placeOrderResult = app.getController('COPlaceOrder').Start();  **if** (placeOrderResult.error) {  start({  PlaceOrderError: placeOrderResult.PlaceOrderError  });  } **else** **if** (placeOrderResult.order\_created) {  showConfirmation(placeOrderResult.Order);  }**else** **if**(placeOrderResult.process3DRedirection){  var jwtUtil = require('int\_cybersource/cartridge/scripts/cardinal/JWTBuilder');  var cardinalUtil = require('int\_cybersource/cartridge/scripts/cardinal/CardinalUtils');  var OrderObject = cardinalUtil.getOrderObject(cart);  var orderdetailsObject = cardinalUtil.getOrderDetailsObject(placeOrderResult.Order,placeOrderResult.authenticationTransactionID);  OrderObject.setOrderDetails(orderdetailsObject);  var jwtToken = jwtUtil.generateTokenWithKey(OrderObject);    var orderstring = JSON.stringify(OrderObject);    app.getView({Order: placeOrderResult.Order,  AcsURL:placeOrderResult.AcsURL,  PAReq:placeOrderResult.PAReq,  PAXID: placeOrderResult.PAXID,  authenticationTransactionID : placeOrderResult.authenticationTransactionID,  jwtToken:jwtToken,  orderstring :orderstring  }).render('cart/cardinalpayerauthentication');  }  } |

##### Update start Function

Update start function to send jwt and order object

function start(context) {

var cart = Cart.get();

// Checks whether all payment methods are still applicable. Recalculates all existing non-gift certificate payment

// instrument totals according to redeemed gift certificates or additional discounts granted through coupon

// redemptions on this page.

var COBilling = app.getController('COBilling');

var CybersourceConstants = require('int\_cybersource/cartridge/scripts/utils/CybersourceConstants');

if (!COBilling.ValidatePayment(cart)) {

COBilling.Start();

return;

} else {

Transaction.wrap(function () {

cart.calculate();

});

Transaction.wrap(function () {

if (!cart.calculatePaymentTransactionTotal()) {

COBilling.Start();

}

});

|  |
| --- |
| COBilling.Start();  }  });  **var** order = "";  **var** jwtToken ="";  **var** iscardinal = **false**;  **var** selectedPaymentMethod = cart.getPaymentInstruments()[0].paymentMethod;  **if**(selectedPaymentMethod.equals(CybersourceConstants.METHOD\_CREDIT\_CARD) || selectedPaymentMethod.equals(CybersourceConstants.METHOD\_SA\_SILENTPOST)  || selectedPaymentMethod.equals(CybersourceConstants.METHOD\_VISA\_CHECKOUT))  {  **var** jwtUtil = require('int\_cybersource/cartridge/scripts/cardinal/JWTBuilder');  **var** cardinalUtil = require('int\_cybersource/cartridge/scripts/cardinal/CardinalUtils');  jwtToken = jwtUtil.generateTokenWithKey();  **var** OrderObject = cardinalUtil.getOrderObject(cart);  order = JSON.stringify(OrderObject);  iscardinal = **true**;  }    **var** pageMeta = require('~/cartridge/scripts/meta');  **var** viewContext = require('app\_storefront\_core/cartridge/scripts/common/extend').immutable(context, {  Basket: cart.object,  jwtToken: jwtToken,  order : order,  iscardinal : iscardinal  });  pageMeta.update({pageTitle: Resource.msg('summary.meta.pagetitle', 'checkout', 'SiteGenesis Checkout')});  app.getView(viewContext).render('checkout/summary/summary');}} |

# Change Logs

|  |  |  |
| --- | --- | --- |
| Action | File Name | Description |
| A | int\_cybersource/cartridge/scripts/cardinal/AccountObject.ds | prototype object which is useful for creating order Object which is passed to cardinal script |
| A | int\_cybersource/cartridge/scripts/cardinal/Billing\_Shipping\_AddressObject.ds | prototype object which is useful for creating order Object which is passed to cardinal script |
| A | int\_cybersource/cartridge/scripts/cardinal/CardinalUtils.ds | contains functions which are used to create order Object which is passed to cardinal script |
| A | int\_cybersource/cartridge/scripts/cardinal/CartItemObject.ds | prototype object which is useful for creating order Object which is passed to cardinal script |
| A | int\_cybersource/cartridge/scripts/cardinal/ConsumerObject.ds | prototype object which is useful for creating order Object which is passed to cardinal script |
| A | int\_cybersource/cartridge/scripts/cardinal/JWTBuilder.ds | contains functions which create JWT Tokens required for cardinal script initialization |
| A | int\_cybersource/cartridge/scripts/cardinal/OrderDetailsObject.ds | prototype object which is useful for creating order Object which is passed to cardinal script |
| A | int\_cybersource/cartridge/scripts/cardinal/OrderObject.ds | prototype object which is useful for creating order Object which is passed to cardinal script |
| A | int\_cybersource/cartridge/scripts/cardinal/TokenObject.ds | prototype object which is useful for creating order Object which is passed to cardinal script |
| M | int\_cybersource/cartridge/scripts/cybersource/libCybersource.ds | 1. getPartnerSolutionID changed to return new one 2. methods added to get custompreference values required for creating JWT Token 3. addPayerAuthEnrollInfo method modified to add extra payerAuthEnrollService input parameters referenceID,mobilePhone,transactionMode. 4. addPayerAuthValidateInfo method modified to add extra payerAuthValidateService input parameter authenticationTransactionID. 5. addPayerAuthReplyInfo method modified to add paSpecificationVersion , directoryServerTransactionID input parameters to ccAuthService request. 6. getTransactionMode method added to return transactionmode value based on device type. |
| M | int\_cybersource/cartridge/scripts/façade/CardFacade.ds | 1. CCAuthRequest method modified to pass missing parameters required for ccAuthService to addPayerAuthReplyInfo method. 2. PayerAuthEnrollCheck modified to pass phone number required for payerauthenrollservice to addPayerAuthEnrollInfo method and new output parameters from payerAuthEnrollReply (authenticationTransactionID,specificationVersion,directoryServerTransactionID,CAVV,UCAFAuthenticationData) response are set to responseObject. 3. In PayerAuthValidation method, null check is added for PaRes value and processorTransactionId is passed to addPayerAuthValidateInfo method and new output parameters from payerAuthValidateReply response (specificationVersion,directoryServerTransactionID) are set to responseObject. |
| M | int\_cybersource/cartridge/scripts/secureacceptance/helper/SecureAcceptanceHelper.js | 1.CreateRequestData method modified for proper functioning of secure acceptance payment. 2. In AuthorizeCreditCard method, authenticationTransactionID from payer enroll service session.privacy.process3DRequestParent set to true are forwarded for 3DS 2.0 Redirection . |
| M | int\_cybersource/cartridge/scripts/visacheckout/façade/VisaCheckoutFacade.js | 1. In PayerAuthEnrollCCAuthRequest method authenticationTransactionID from payerAuthEnrollReply response is set to responseObject required for cardinal script continue function. 2. processorTransactionId from payments.validated event in cardinal script is passed to payerauthvalidation service and null check for PaRes value in PayerAuthValidationCCAuthRequest method. |
| M | int\_cybersource/cartridge/scripts/visacheckout/helper/VisaCheckoutHelper.js | processorTransactionId from payments.validated event in cardinal script is passed to PayerAuthValidationCCAuthRequest method in payerAuthValidation method. |
| A | int\_cybersource/cartridge/templates/default/cardinal/songbird.isml | This isml contains script and required html dom elements and cardinal events configured for initialising the cardinal script. |
| A | int\_cybersource/cartridge/templates/default/cart/cardinalpayerauthentication.isml | This isml contains script and required html dom elements and cardinal events configured for initialising the cardinal script and cardinal continue function to start a transaction. |
| M | int\_cybersource/cartridge/templates/default/secureacceptance/secureAcceptanceIframeSummary.isml | cybersource-custom.js script file is added which opens an SA-Iframe in the summary page. |
| M | int\_cybersource/cartridge/webreferences/CyberSourceTransaction.wsdl | wsdl properties modified for updating the cybersource endpoint. |
| A | int\_cybersource/cartridge/webreferences/CyberSourceTransaction\_1.155.xsd | new XSD file for updated cybersource endpoint |
| M | int\_cybersource\_controllers/cartridge/scripts/Cybersource.js | In Process3DRequestParent method, processorTransactionId returned by cardinal commerce script is fetched from httpParameterMap in the request received and an empty check is added for it and it is passed to payerauthvalidation services based on the paymentinstrument. |
| A | configuration/PayerAuth-Meta.xml |  |

# Support French Processor

To support French Processor, follow one of the below steps:

* + - 1. Install the latest Cybersource cartridge from Salesforce Marketplace – This is recommended approach **(OR)**
      2. Manually add below changes (SFCC’s GitHub commit) into your workspace code. It is always recommended to test (both 3ds2 and 3ds1 test cards) the changes in your lower box before promoting to Production org.

<https://github.com/SalesforceCommerceCloud/link_cybersource/commit/6f0b66980ce7e60ffb135b65b94d173fe6027c64>

<https://github.com/SalesforceCommerceCloud/link_cybersource/commit/97d2fc51084bb9a88b66101e75b5bbc91329ba14>

<https://github.com/SalesforceCommerceCloud/link_cybersource/commit/dc7dc3ef176673b1247b49d2f0e54fbe90d2be8b>

And update base cartridge with below code.

Update the base cartridge file name [app\_storefront\_core/cartridge/js/pages/checkout/billing.js](https://github.com/krishnaprasadsn25/cybersource-plugins-salesforce-1/commit/f4e54f960f96c50dfee534742cc4a2e02e0ed21c#diff-9408dda1551401fcdf8e3b81b06f17aa107ed26d93694b2f0e14163429d6cbbc)

|  |
| --- |
| cctoken : cctoken,  ccnumber : ccnumber,  cvn : cvn,  month : month,  expyear : expyear,  cctype : cctype,  format : 'ajax' |

Update the base cartridge file name: [app\_storefront\_core/cartridge/static/default/js/app.js](https://github.com/krishnaprasadsn25/cybersource-plugins-salesforce-1/commit/f4e54f960f96c50dfee534742cc4a2e02e0ed21c#diff-8796c34fcce9816e9dc28d91e3170d2780a321b8b1a0366a74c446fe214c7fd0)

|  |
| --- |
| cctoken : cctoken,  ccnumber : ccnumber,  cvn : cvn,  month : month,  expyear : expyear,  cctype : cctype,  format : 'ajax' |

Update base cartridge file [app\_storefront\_core/cartridge/templates/default/account/payment/paymentinstrumentdetails.isml](https://github.com/krishnaprasadsn25/cybersource-plugins-salesforce-1/commit/f75b1456fb03b27fa663db266bc98362a70d24d7#diff-790538e75d366b1405f64daf55c84af283f693e54e4d2e9d18fb18f27bd23edc)

|  |
| --- |
| <isinputfield formfield="${pdict.CurrentForms.paymentinstruments.creditcards.newcreditcard.number}" dynamicname="true" type="input" attributes="${numberAttributes}"/>  <isinputfield formfield="${pdict.CurrentForms.paymentinstruments.creditcards.newcreditcard.cvn}" dynamicname="true" type="input" attributes="${cvnAttributes}"/> |