Guide to Upgrade Cybersource Microform to v2 in SAP Commerce B2C

Steps to Upgrade Microform to v2

1. Update the FlexMicroformController.java

File Path:
hybris/bin/b2c/isvb2cpaymentaddon/acceleratoraddon/web/src/isv/sap/payment/addon/b2c/controllers/pages/checkout/payment/flex/FlexMicroformController.java

The FlexMicroformController is responsible for generating the capture context and verifying the token. This step updates the controller to support Microform v2.

Changes Made:

Generate Capture Context:

- Updated the newJwk method to include clientLibrary and clientLibraryIntegrity values, which are required for loading the Microform v2 JavaScript library.

Updated Code:

Add Import:

```
import isv.sap.payment.addon.utils.AjaxResponse;
import org.apache.commons.text.StringEscapeUtils;
```

Replace the function:

```
@GetMapping(value = "/newJwk", produces = MediaType.APPLICATION_JSON_VALUE)
@ResponseBody
public AjaxResponse newJwk(final HttpSession session, final UriComponentsBuilder
uriComponentsBuilder) {
  final String targetOrigin = uriComponentsBuilder
        .replacePath(null).replaceQuery(null).userInfo(null).fragment(null)
        .build()
        .toUriString();
  final Map<String, String> captureContext = flexService.createKey(targetOrigin);
session.setAttribute(FLEX_CAPTURE_CONTEXT_ATTRIBUTE,captureContext.get("capture
Context"));
   return AjaxResponse.success()
         .put("captureContext",
StringEscapeUtils.escapeHtml4(captureContext.get("captureContext")))
         .put("clientLibrary",
StringEscapeUtils.escapeHtml4(captureContext.get("clientLibrary")))
         .put("clientLibraryIntegrity",
StringEscapeUtils.escapeHtml4(captureContext.get("clientLibraryIntegrity")));
}
```

2. Decode Capture Context

The decoding of the capture context is handled within the DefaultFlexService class.

Generate Capture Context

- The createKey method in the DefaultFlexService class generates the capture context using the Cybersource Microform v2 API. It includes the allowed card networks, target origins, and other required parameters.

Changes Made:

- Replace the current API version: *hybris/bin/b2c/isvpayment/lib/isv-payment-api-3.0.4.*

3. Update the microform.tag

File Path:
hybris/bin/b2c/isvpaymentaddon/acceleratoraddon/web/webroot/WEB-INF/tags/responsive/payment/flex/microform.tag

The microform.tag file is responsible for initializing and loading the Microform iframe on the frontend.

Changes Made:

Load Microform v2 Library:

- Dynamically load the Microform v2 JavaScript library using the clientLibrary and clientLibraryIntegrity values.

Updated Code:

Remove the following attributes:

```
<%@ attribute name="flexSdkUrl" required="true" type="java.lang.String" %>
<script src="${flexSdkUrl}"></script>
```

Update the following functions:

```
setup: function () {
  if (!MICROFORM.loaded) {
    MICROFORM.createNewJwk()
  }
},

loadScript: function loadScript(clientLibrary, clientLibraryIntegrity) {
  return new Promise((resolve, reject) => {
    try {
      let script = document.getElementById('flexClientLibrary');
      if (null == script) {
        let scriptElement = document.createElement("script");
        scriptElement.type = "text/javascript";
        scriptElement.src = clientLibrary;
        scriptElement.integrity = clientLibraryIntegrity;
        scriptElement.crossOrigin = 'anonymous';
        scriptElement.id = "flexClientLibrary";
        document.body.appendChild(scriptElement);
        scriptElement.onload = () => {
          resolve(true);
        };
        scriptElement.onerror = () => {
                reject(false);
            };
        } else {
            resolve(true);
        }
    } catch (error) {
        reject(false);
    }
  });
},

createNewJwk: function () {
  var created = false;
  MICROFORM.waitBegin();
  $.ajax({
    url: MICROFORM.newJwkEndpointUrl,
    cache: false,
    async: false,
    dataType: 'json',
    success: function (result) {
        MICROFORM.captureContext = result.data.captureContext;
        MICROFORM.loadScript(result.data.clientLibrary,
result.data.clientLibraryIntegrity).then(() => {
            MICROFORM.waitBegin();
            var flex = new Flex(MICROFORM.captureContext);
            var microform = flex.microform({styles: MICROFORM.onGetStyles()});
            var number = microform.createField('number', {placeholder: 'Enter card number'});
            var securityCode = microform.createField('securityCode', {placeholder: '***'});

            securityCode.load(MICROFORM.flexSecurityCodeSelector);
            number.load(MICROFORM.flexCardNumberContainerSelector);

            if (MICROFORM.cardChangeEventHandler) {
                number.on('change', MICROFORM.cardChangeEventHandler);
            }
            if (MICROFORM.cvnChangeEventHandler) {
                securityCode.on('change', MICROFORM.cvnChangeEventHandler);
            }
            MICROFORM.microformInstance = microform;
            MICROFORM.loaded = true;
            MICROFORM.waitComplete();
         }).catch(() => {
            // Handle errors here
         });
         created = true;
      },
      error: function (jqXHR, textStatus) {
         MICROFORM.reportError('Failed to create new JWK', textStatus);
      }
   });

   MICROFORM.waitComplete();
   return created;
},
```

4. Update the JSP View (flexCardPaymentDetails.jsp)

File Path:
hybris/bin/b2c/isvpaymentaddon/acceleratoraddon/web/webroot/WEB-INF/views/responsive/pages/checkout/multi/payment/flexCardPaymentDetails.jsp

The JSP view is responsible for rendering the payment form and including the Microform iframe.

Changes Made:

Include Microform v2 Script:

- Dynamically include the Microform v2 JavaScript library using the clientLibrary and clientLibraryIntegrity values.

Remove the below line:

```
flexSdkUrl="${flexSdkUrl}"
```

5. Update the SummaryCheckoutStepController.java

File Path:
hybris/bin/b2c/isvb2cpaymentaddon/acceleratoraddon/web/src/isv/sap/payment/addon/b2c/controllers/pages/checkout/steps/SummaryCheckoutStepController.java

The SummaryCheckoutStepController is responsible for preparing the checkout summary page.

Changes Made:

Add Microform v2 Attributes:

- Add clientLibrary and clientLibraryIntegrity values to the model.

Updated Code:

Remove the following lines:

```
@Value("${isv.payment.flex.microform.sdk.url}")
private String flexSDKUrl;
```

Update function:

```
protected void prepareFlexMicroformData(final Model model) {
   if (checkoutPciStrategy.getSubscriptionPciOption().equals(FLEX)) {
      model.addAttribute("flexCardTypeSelection", flexCardTypeSelection);
   }
}
```

6. Update Configuration Properties (project.properties)

File Path:
hybris/bin/b2c/isvpayment/project.properties

The project.properties file contains the configuration for Microform v2.

Changes Made:

Add Allowed Card Types:

Add the following configuration to specify the allowed card types:

```
isv.payment.network.to.allowedCardType=<Allowed Card Types>
```

Possible Values:
VISA,MAESTRO,MASTERCARD,AMEX,DISCOVER,DINERSCLUB,JCB,CUP,CARTESBANCAIRES