CyberSource Cartridge

Version 1.0



Contents

Summary 5

Component Overview 6

Functional Overview 6

Credit Card Authorization Service 6

Taxes 7

Address Verification Service (AVS) 8

Delivery Address Verification Service (DAV) 8

Bill Me Later (BML) 8

Decision Manager 9

Use Cases Scenarios 10

Credit Card Authorization 10

Taxes 12

Address Validation Service (AVS) 13

Delivery Address Verification Service (DAV) 13

BML 14

Decision Manager 14

Limitations, Constraints 15

Compatibility 15

Implementation Guide 16

Custom Code 16

Payment Services 16

Tax Service 16

Address Verification Service 17

Delivery Address Validation Service 17

Site Configuration 18

Import Meta Data 18

Configure Site Preferences 18

Applying CyberSource Cartridge to the Site 20

Testing 20

Authorize Credit Card 21

Tax Service 21

Address Verification Service (AVS) 21

Delivery Address Verification Service (DAV) 21

Cartridges Structure and Reference 22

Pipelines 22

Scripts 22

Templates 23

Configuration Files 23

Typical Project Plan 24

Roles, Responsibilities 24

Typical Efforts and Timelines 24

CyberSource Site Preferences 26

Known Issues 28

CyberSource document links 29

Release History 30

# Summary

This document provides technical overview and implementation details for each CyberSource service integrated within Demandware platform. The CyberSource cartridge extends the functionality of Demandware Storefront, enabling real time access to CyberSource eCommerce transaction services listed below.

* Credit Card Authorization – The CyberSource Credit Card Authorization service provides merchant with a mechanism to get authorization for the order amount. The authorization service validates the card based and authorize card for the order amount.
* CyberSource Address Verification (AVS) – The CyberSource AVS service provides merchants with a mechanism to reduce merchant banking fees, by verifying billing information before authorizing payment for customer purchases. Although the AVS service is automatically called during the authorization process, the behavior resulting from specific AVS responses and its interaction with the payment process is customizable through storefront configuration.
* Tax Service – The CyberSource tax calculation service provides merchants with a complete tax calculation service according to and pursuant to domestic and international tax regulations, including but not limited to, district, city, county and state levels of governing tax authority.
* Delivery Address Verification (DAV) Service – The CyberSource DAV service provides merchants with an optional mechanism to prevent, limit or correct faulty shipping information, related to improperly entered or formatted information from the customer. This service helps reduce the potential additional costs resulting from undeliverable or returned merchandise.
* Bill Me Later Service – The Bill Me Later service from Bill Me Later, Inc., and available via your single connection to CyberSource, allows your customers to make purchases using an instant line of credit.
* Decision Manager – CyberSource Decision Manager Service provides set of tools that merchant to evaluate rules and chose tool and return a decision of “Accept”, “Reject”, or “Review”. Merchant can also setup process to ignore certain rules when necessary.

# Component Overview

## Functional Overview

### Credit Card Authorization Service

The credit card authorization service pipeline allows storefront application to request for credit authorization for the total order amount. The pipeline makes the credit card authorization web service call to CyberSource authorization service and receive confirmation about the availability of the funds.

The Demandware Cybersource–AuthorizeCreditCard pipeline populates the authorization request with ship-to, bill-to, credit card data, and purchase total data from the basket and invokes the authorization web service call using CyberSource web service API.

Credit Card Authorization sequence flow:

1. Creates CyberSource authorization request using ship-to, bill-to, credit card data, and purchase total data from the current basket.
2. If authorize Payer is configured, then make the authorize payer request, if not ignore and continue with the authorization request.
3. Create credit card authorization request.
4. If DAV is enabled, then set up DAV business rules, as needed.
5. Set up AVS Ignore Result business rule for request with AVS Ignore Flags specification, as needed.
6. Make actual service call to CyberSource Simple Order API.
7. If Delivery Address Verification is enabled, then:
   1. Capture pertinent DAV result information & DAV Reason Code
   2. If DAV fails and DAV On Failure is set to ‘REJECT’, then exit immediately with rejection response
8. If DAV On Failure is set to ‘APPROVE’ and the DAV Reason Code is a fail code (not 100), then:
   1. Exit immediately with declined or review response, as merchant defines
9. Capture pertinent AVS information
10. Validate authorization reason code and set corresponding end node, based on auth response code.

The list of activities depicted in the following diagram take place when API request is made for an online credit card authorization: [**Source, CyberSource Credit Card Service, October 2009**]

**Figure 1** Processing an Online Authorization [**Source, CyberSource Credit Card Service, October 2009**]



**1**The customer places an order and provides the credit card number, the card expiration date, and other information about the card.

2   You send a request for authorization over a secure Internet connection. If the customer buys a digitally delivered product or service, you can request both the authorization and the capture at the same time. If the customer buys a physically fulfilled product, do not request the capture until you ship the product.

3   CyberSource validates the order information, then contacts your payment processor and requests authorization.

4   The processor sends the transaction to the card association, which routes it to the issuing bank for the customer’s credit card. Some card companies, including Discover and American Express, act as their own issuing banks.

5   The issuing bank approves or declines the request. Depending on the card type, the bank could also use the Address Verification Service (AVS) to determine whether the customer provided the correct billing address. For more information about AVS, refer to AVS service documents via the CyberSource Services Documentation at <http://www.cybersource.com/support_center/support_documentation/services_documentation/payment.php> or as described in this integration guide.

6   CyberSource runs its own tests, then tells you if the authorization succeeded.

7   Response is sent back to the client.

### Taxes

Online Customer adds Product(s) to Cart and proceeds to Checkout.

As soon as shipping information is entered and validated, taxes are updated to reflect current tax rates based on six basic criteria:

1. Customer ship to address
2. Merchant ship from address
3. Merchant point of order origin (POO)
4. Merchant point of order acceptance (POA)
5. Product code
6. Merchant nexus

Product information is provided on an individual line item basis and all merchant/request IDs are captured for future reference. When the customer enters in shipping information, the Tax Service is called to calculate taxes.

### Address Verification Service (AVS)

AVS does not exist as a stand-alone callable service. Please refer to the Credit Card Authorization Service walkthrough for high level walkthrough.

### Delivery Address Verification Service (DAV)

DAV service may be run as a stand-alone callable service, as well as be performed as a part of other services. Please refer to Credit Card Authorization Service for more information regarding the DAV service, as an integral part of payment auth.

As a stand-alone service, the process is defined as:

* Customer enters shipping information
* Shipping information passes client-side validation (required elements filled in)
* Shipping information passes basic server-side validation (syntactically correct)
* Request is made to CyberSource DAV Service
* Response returns DAVReasonCode (100=Success)
* End node returns either: authorized, declined or error (authorized==success, declined==failure)
* Captured validation information is extracted from pipeline dictionary to present user with choices to correct problems, confirm “standardized” formatting or try again
* If service is successful, allow ShippingAddress save operation to continue

### Bill Me Later (BML)

A customer selects Bill Me Later during the checkout process at your site, similar to selecting the option to pay via Visa or MasterCard.

To request a Bill Me Later authorization, Demandware pipeline sends a request for a credit card authorization but instead of including a credit card number in the request, send the customer’s Bill Me Later account number set in the custom preferences.

To bill the customer, send a request for a credit card capture. No additional capture request fields are required for a Bill Me Later capture, unless you are processing multiple captures.

The Bill Me Later authorization service pipeline allows storefront application to request for credit authorization for the total order amount through the Bill Me Later.

The Demandware Cybersource–AuthorizeBML pipeline populates the authorization request with ship-to, bill-to, credit card data, and purchase total data from the basket and invokes the authorization web service call using CyberSource web service API.

Bill Me Later authorization sequence process:

* If it is the first time a customer has used Bill Me Later, they are presented with terms and conditions and asked for date of birth and last 4 digits of social security number.
* Once terms are agreed to, checkout continues as usual.
* A real-time credit decision is made and the consumer is notified within 3-5 seconds.
* On subsequent purchases, the customer simply chooses Bill Me Later, a real-time credit decision is made and the customer is notified within 3-5 seconds.
* In rare circumstances, additional verification steps are applied.
* After completion of the first purchase a welcome email is sent by Bill Me Later. This email contains a secure link for login to the self-service website, as well as a user ID for the customer.
* Your customer typically receives a bill (sent by Bill Me Later, Inc.) in their mailbox within fifteen days of the settlement posting to the customer's account. The due date for this bill is 25 days from the date the bill is rendered.
* You submit the transaction for settlement upon shipment of goods or rendering the service—just as you do with a credit card purchase. Funding is received within the same timeframe as that of a purchase made with a credit card.

### Decision Manager

The CyberSoruce Decision Manager provides Merchant and ability to set business rules, provide case management, and Reporting.

The CyberSource Decision Manager Business rule engine allows Merchant to analyze the order data based on predefined or custom rules. The business rules can be set by orders, by category, or by SKU. The

The Demandware CyberSource Cartridge pipeline processes incoming Decision Manager Request and set contained orders to the new decision made through the decision manager.

Entry point into Demandware pipeline:

http://<sandboxname>/on/demandware.store/Sites-<store>-Site/default/Cybersource-NewDecision?content=<xml content>

Sample incoming reviewed order status update xml file

<?xml%20version="1.0"%20encoding="utf-8"?>

<!DOCTYPE CaseManagementOrderStatus SYSTEM "https://ebctest.cybersource.com/ebctest/reports/dtd/cmorderstatus\_1\_1.dtd">

<CaseManagementOrderStatus

MerchantID="sample\_merchant"

Name="Case Management Order Status"

Date="2008-12-18 12:22:09 GMT"

Version="1.1"

nxmlns="http://reports.cybersource.com/reports/cmos/1.0">

<Update MerchantReferenceNumber="10679256010963322294714" RequestID="1744185012770167904567">

<OriginalDecision>REVIEW</OriginalDecision>

<NewDecision>ACCEPT</NewDecision>

<Reviewer>sample\_reviewer</Reviewer>

<ReviewerComments>sample\_comment</ReviewerComments>

<Queue>sample\_queue</Queue>

<Profile>test</Profile>

<FollowonResult>

<Status>Success</Status>

<Application>Credit%20Card%20Settlement</Application>

<RequestID>1744185012770167904567</RequestID>

<Decision>Accept</Decision>

<ReasonCode>100</ReasonCode>

<RCode>1</RCode>

<RFlag>SOK</RFlag>

<RMsg>Request%20was%20processed%20successfully.</RMsg>

</FollowonResult>

</Update>

</CaseManagementOrderStatus>

## Use Cases Scenarios

### Credit Card Authorization

The following table outlines the possible Demandware actions based on the response of the CyberSource gateway. Each client may choose to handle the response code differently. As of release 2.10, all errors logged as "fatal", can activate an email alert to recipients identified in business manager.

|  |  |  |  |
| --- | --- | --- | --- |
| **Response** | **DW Storefront Action** | **Cyber-**  **Source Code** | **CyberSource suggested response** |
|  |  |  |  |
| Successful transaction. | Continue Checkout | 100 |  |
|  |  |  |  |
| **Validation Errors** | | | |
| Request is missing one or more fields | Should not occur as validation should catch this Show user "denied" error message Log fatal error (email alert) | 101 | See the reply fields missingField\_0...N for which fields are missing. Resend the request with the complete information. |
|  |  |  |  |
|  |  |  |  |
| One or more fields in the request contains invalid data. | Should not occur as validation should catch this Show user "denied" error message Log fatal error (email alert) | 102 | See the reply fields invalidField\_0...N for which fields are invalid. Resend the request with the correct information. |
|  |  |  |  |
| **System Errors** | | | |
| General system failure. | Show user "Unable to process - Call Cust Service" error message Log fatal error (email alert) | 150 | Wait a few minutes and resend the request. |
|  |  |  |  |
|  |  |  |  |
| The request was received but there was a server time-out. | Show user "Unable to process - Call Cust Service" error message Log fatal error (email alert) | 151 | Wait a few minutes and resend the request. |
|  |  |  |  |
|  |  |  |  |
| The request was received but there was a service time-out. | Show user "Unable to process - Call Cust Service" error message Log fatal error (email alert) | 152 | Wait a few minutes and resend the request. |
|  |  |  |  |
| The request just wait and then timeout, ends up as exception on the Demandware script | This could be one of the unique scenario where CyberSource waits for the Merchant’s bank to authorize the order and exceeds timeout sets at the Demandware. This ends up into SOAP exception. Client code can handle this scenario differently. | Script sets Rason Code to 999 | Handle at client’s end depending on business rules associated with this scenario. |
|  |  |  |  |
| **Authorization denied errors** | | | |
| Declined the request because the card has expired. | show user "Auth denied" error message | 202 | Request a different card or another form of payment. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| The account number is invalid. | show user "Auth denied" error message | 231 | Request a different card or other form of payment. |
|  |  |  |  |
| **Gateway Account problem** | | | |
| There is a problem with your merchant configuration. | Show user "Unable to process - Call Cust Service" error message Log fatal error (email alert) | 234 | Do not resend the request. Contact Customer Support to correct the configuration problem. |
|  |  |  |  |
| **Fraud Management** | | | |
| The fraud score exceeds your threshold. | Show user "Unable to process - Call Cust Service" error message Log fatal error (email alert) | 400 |  |
|  |  |  |  |
| The order is marked for review by Decision Manager. | Proceed with checkout Leave DW order "unconfirmed" | 480 |  |
|  |  |  |  |
| The order is rejected by Decision Manager. | Show user "Unable to process - Call Cust Service" error message Log fatal error (email alert) | 481 |  |

### Taxes

List of use cases and appropriate action taken listed below:

|  |  |
| --- | --- |
| **Use case scenarios** | **Result** |
| If shipping information is specified, then a request is made to the Tax Service | If successful, the contents of the Basket are taxed and price totals are adjusted.  If failed, because of service outage or failed address verification then don’t update anything. Other services must handle AVS/DAV/Service outages before successful checkout and final sales tax calculation. Failure is logged for email notification. |

### Address Validation Service (AVS)

Note that AVS does not run as an independent process, but is instead an optional, integrated aspect of payment authorization. List of use cases and appropriate action taken listed below:

|  |  |
| --- | --- |
| **Use case scenarios** | **Result** |
| AVS Ignore Result set to true | AVS Information is captured, but does not affect authorization response. |
| AVS Ignore Result set to false | AVS information is captured and if result from AVS is error or declined, then propagate that result up to the calling service. |
| AVS Ignore Result is set to false & AVS Decline Flags is defined | Seed request with additional result codes which should also result in a declined response. |

### Delivery Address Verification Service (DAV)

List of use cases and appropriate action taken listed below:

|  |  |
| --- | --- |
| **Use case scenarios** | **Result** |
| DAV Enable is set to false | No DAV information will be requested. No correction/validation information will be collected from the response. |
| DAV Enable is set to true, DAV On Failure set to REJECT | DAV information will be requested from the calling service. DAV related corrections and validation information is captured, and a DAV-related failure will be propagated to the calling service. |
| DAV Enable is set to true, DAV On Failure set to APPROVE | DAV information will be requested from the calling service. DAV related corrections and validation information is captured, but the result does not affect Authorization result. |

### BML

List of use cases and appropriate action taken listed below:

|  |  |
| --- | --- |
| **Use case scenarios** | **Result** |
| BML Authorization failed with response.decision = ERROR | Client application to display appropriate user friendly message to the end user. |
| BML Authorization failed with response.decision = ACCEPT | Pipelien sets the Authorization code to BMLPaymentInstrument.paymentTransaction.transactionID and ends with Authorized status |
| BML Authorization failed with script error, or exception | Pipeline ends with error status, client code to display appropriate error message to the end user. |

### Decision Manager

Updates order status with the new decision set through the Decision Manager. The order status is updated in Demandware through the incoming xml. There are following possible options:

The new order status can be set to either accepted or rejected. TheCybersource-NewDecision pipeline retrieves the order for the incoming XML content, read order number from the XML, and updates corresponding storefront order with the status passed in the XML for the order.

List of use cases and appropriate action taken listed below:

|  |  |
| --- | --- |
| **Use case scenarios** | **Result** |
| Incoming order status is set to “ACCEPT” | Read order from the order table; Update the status and set that add it to the accepted orders collection.  The accepted order collection can then be used to log and alert.  Sets the HTTP response code “200”. |
| Incoming order status is set to “REJECT” | Read order from the order table; update the status and set that add it to the declined orders collection.  The declined order collection can then be used to log and alert.  Sets the HTTP response code “200”. |

## Limitations, Constraints

Not currently implemented:

* Multiple shipments. Tax rates are only calculated for a single shipment per order. To implement tax service calculation for multiple shipments, a separate web service call must be made for each distinct “ship to” location.
* Custom User Interface components to correct address validation (DAV/AVS) errors and/or omissions or to confirm “standardized” address format corrections. All pertinent data is collected, but because each merchant will have customized specifications how to deal with such information (or use other 3rd party solutions to play the same role), no default user interface is provided.

## Compatibility

This cartridge is supported under Demandware API release 2.10.0 and onward.

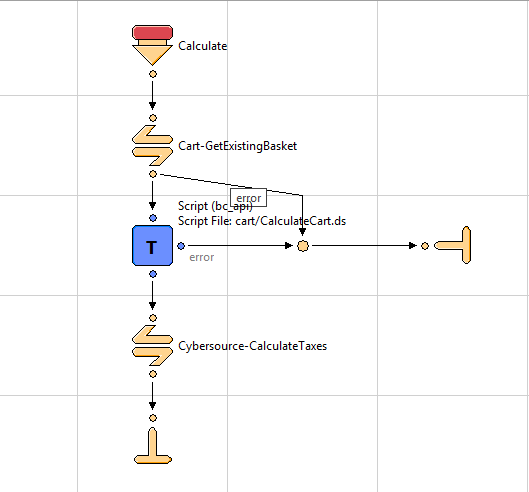
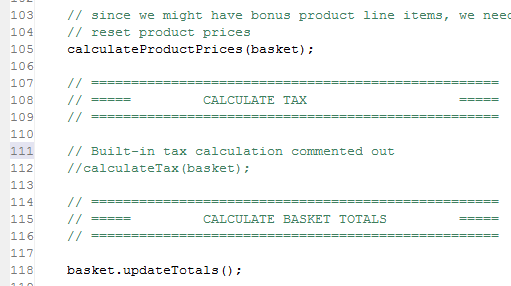
# Implementation Guide

## Custom Code

### Payment Services

1. Update COPlaceOrder-HandlePayments pipeline to include Cybersource-AuthorizeCreditCard.
2. Add logic to handle following responses from the Cybersource-AuthorizeCreditCard:
3. Authorized
4. Error
5. Declined and
6. Review

### Tax Service

1. Update Cart-Calculate pipeline to run the Cybersource-Taxes call node after running the cart/calculateCart.ds script.  
   
2. Comment out the built in tax calculation function call in cart/calculateCart.ds  
   
3. Provide Site Preference values for the Ship From, POO, POA & Nexus related fields.

### Address Verification Service

1. Provide Site Preference values for 2 AVS-related business rules:
   1. CsAvsIgnoreResult – Determines whether AVS failures will force an auth failure
   2. CsAvsDeclineFlags –Determines how “correct” an address must be to produce a failure result
2. Augment UI interaction nodes to deal with AVS failure or correction confirmation dialogs, wherever Payment Authorization takes place, typically within the COPlaceOrder-Submit pipeline.

### Delivery Address Validation Service

1. Provide Site Preference values for 2 DAV-related business rules:
   1. CsDavEnable – Determines whether DAV features are enabled for payment auth requests
   2. CsDavOnAddressVerificationFailure –Determines whether a DAV failure will result in a payment auth failure
2. Augment UI interaction nodes to deal with AVS failure or correction confirmation dialogs, wherever Payment Authorization takes place, typically within the COPlaceOrder-Submit pipeline.

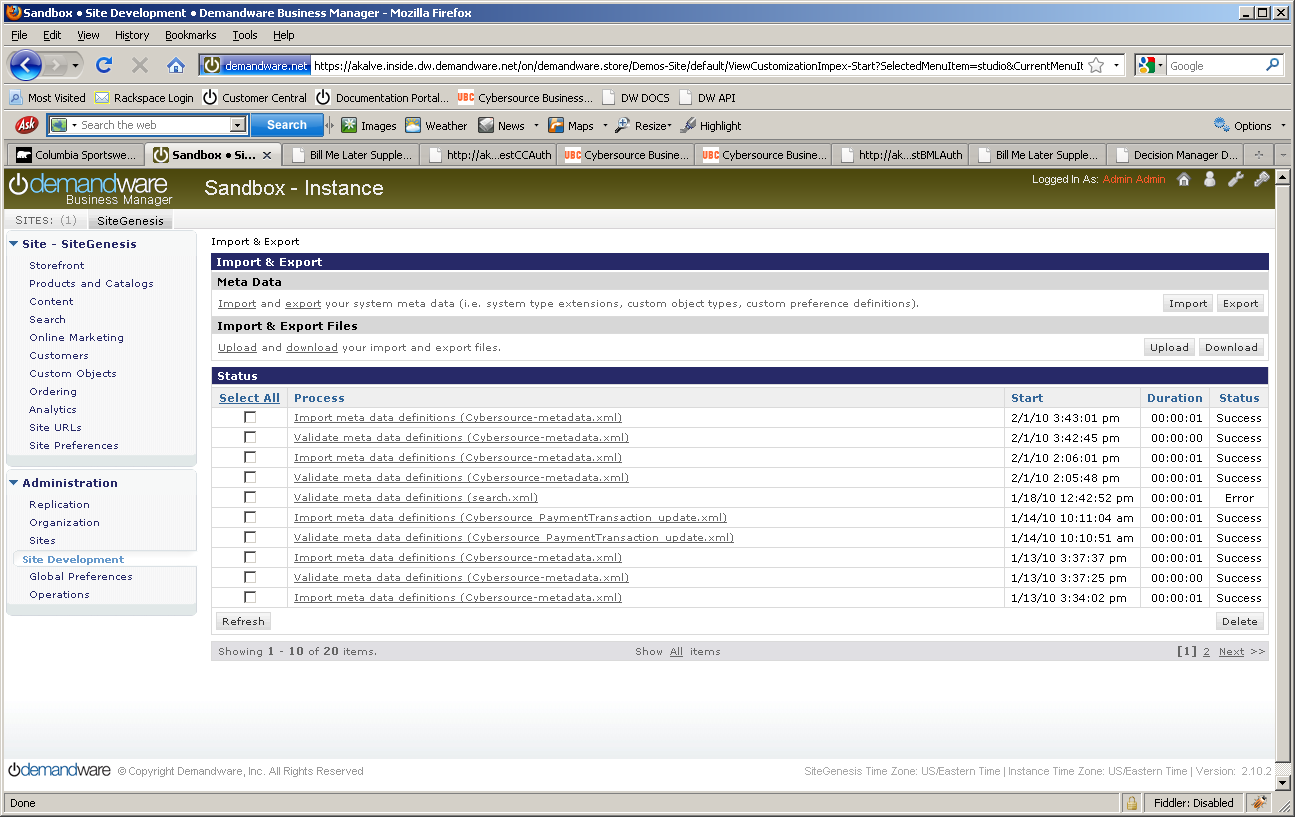
## Site Configuration

### Import Meta Data

Import following site configuration meta-data through Business Manager:

* /int\_cybersource/configuration/Cybersource-metadata.xml – sets all the site preferences
* /int\_cybersource/configuration/Cybersource\_PaymementTransaction\_updates.xml – add custom attributes to the Payment transaction object.

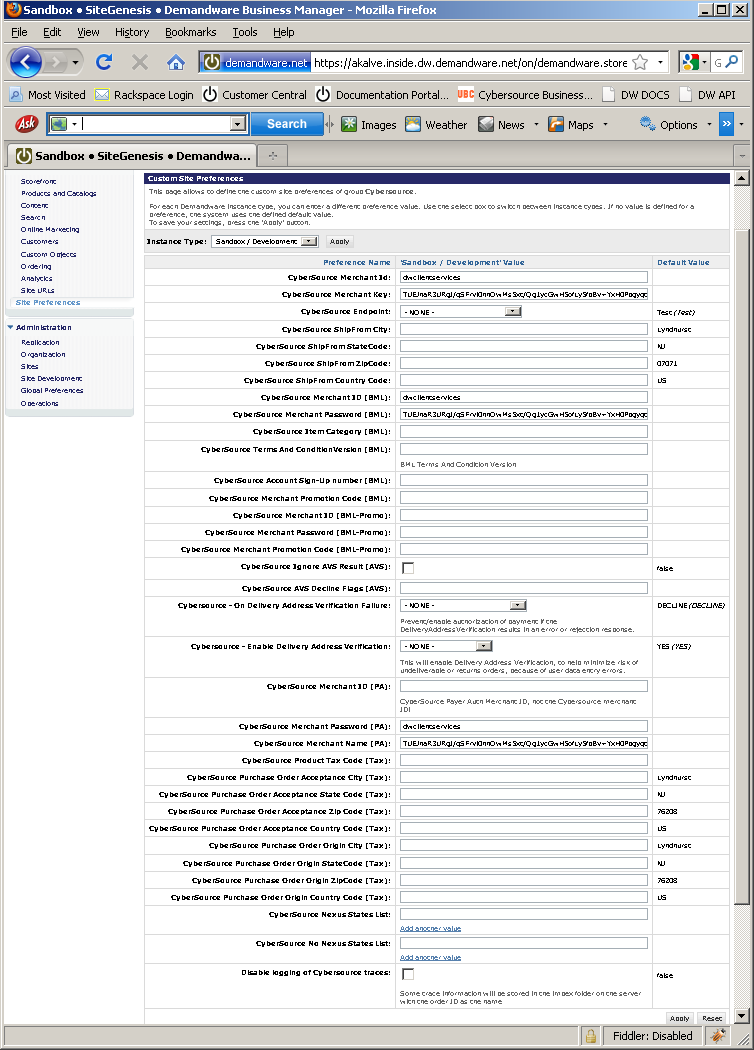
The following Business Manager script shot depicts the import / Export functionality:



### Configure Site Preferences

Update CyberSource site preference through Business Manager > StoreFront Site> Site Preferences.

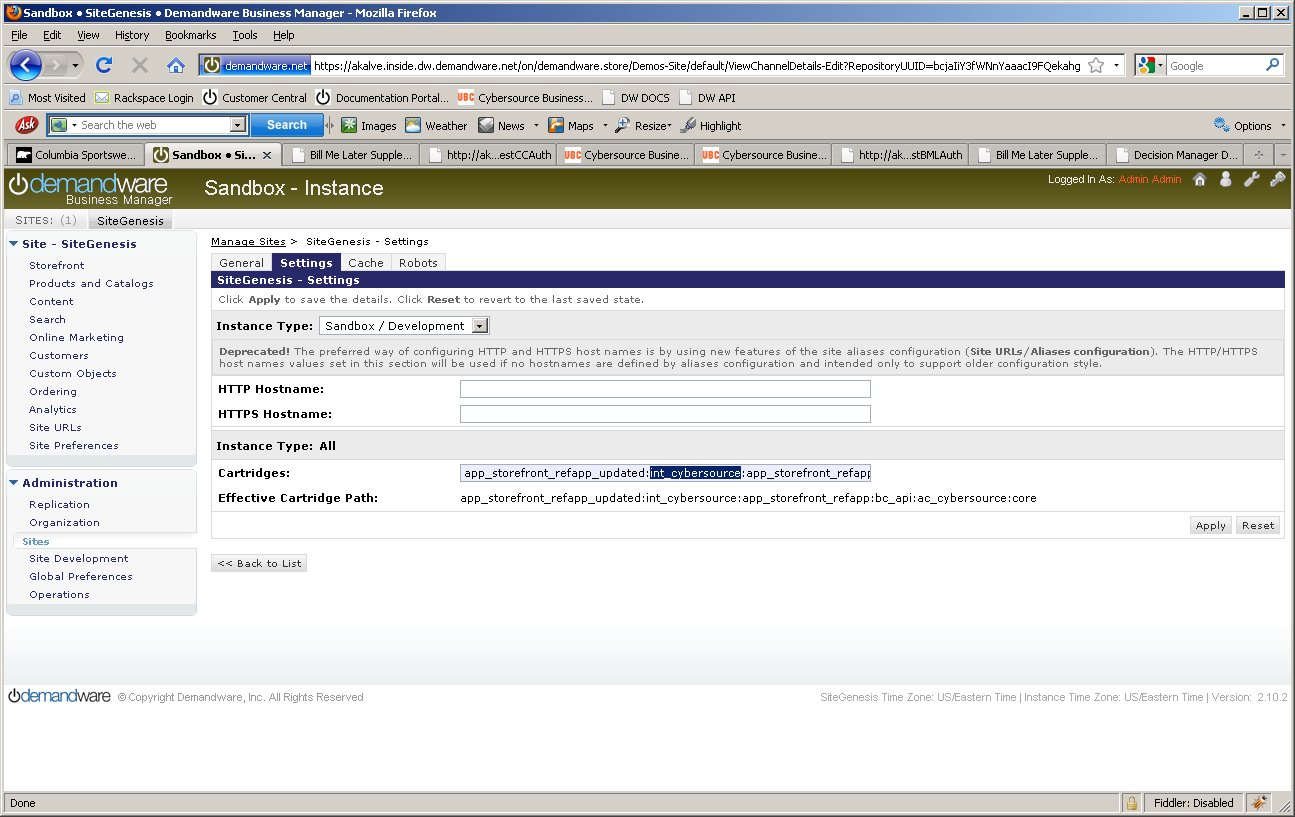
The screen shot below depicts the site preferences configuration:



### Applying CyberSource Cartridge to the Site

Go to the “Administration” in the left hand list to expand the menu and select Sites > Manage Sites link. This will open a list of the active sites on the Demandware platform in your account. Click on the site for which you wish to add the CyberSource cartridge. This will open the General Settings page for that site.

Add int\_cybersource cartridge to the cartridge path as depicted in the following screen:



## Testing

Use CybersourceUnitTest pipeline to test all the services as follows:

### Authorize Credit Card

Use and modify the CybersourceUnitTest-TestCCAuth pipeline and associated scripts and sub-pipelines. The end node of the unit test pipeline is a template which displays all relevant request/response information in an easy to digest manner. User can change static credit card and address data to observe various responses.

### Tax Service

Use and modify the CybersourceUnitTest-TestTaxes pipeline and associated scripts and sub-pipelines. The script nodes for creating CreateMockShipTo and CreateMockBillTo objects have bindings to produce invalid and missing fields, but otherwise can be manually modified to test against any domestic or international address.

The end node of the unit test pipeline for taxes is a template which displays all relevant request/response information in an easy to digest manner, to aid the debugging the various response codes and corrected address response.

### Address Verification Service (AVS)

Use and modify the CybersourceUnitTest-TestCCAuth pipeline and associated scripts and sub-pipelines. By running simplified payment authorizations with different site preferences set, you can see how the AVS process works and how that result affects the overall payment authorization process.

### Delivery Address Verification Service (DAV)

Use and modify the CybersourceUnitTest-TestCCAuth pipeline and associated scripts and sub-pipelines. By running simplified payment authorizations with different site preferences set, you can see how the DAV process works and how that result affects the overall payment authorization process.

To test the stand-alone DAV service, use and/or modify the CybersourceUnitTest-TestDAVCheck pipeline and associated scripts and sub-pipelines. Like the TestTaxes pipeline, you can customize the mocked data to simulate various situations that need to be handled.

The end node of the unit test pipeline for the stand-alone DAV Service is a template which displays all relevant request/response information in an easy to digest manner, to aid the debugging the various response codes and corrected address response.

# Cartridges Structure and Reference

### Pipelines

##### Cybersource

This contains the nodes that make the actual call to CyberSource based on what is needed, cc auth, bml, tax or address verification.

##### CybersourceData

This contains the node that creates the objects needed for the CyberSource pipeline. Each node in the pipeline creates a needed object for each request. This is the pipeline that will be modified by customers, as each customer may need to pass different information based on their needs

##### CybersourceUnitTesting.

Used for testing.

### Scripts

There is a JS object for each request data elements. List of DS objects used:

* Cybersource\_BillTo\_Object.ds
* Cybersource\_BML\_Object.ds
* Cybersource\_Card\_Object.ds
* Cybersource\_Item\_Object.ds
* Cybersource\_PurchaseTotals\_Object.ds
* Cybersource\_ShipFrom\_Object.ds
* Cybersource\_ShipTo\_Object.ds
* Cybersource\_TaxRequest\_Object.ds

The following scripts are used in CyberSource pipeline to make the CyberSource web service calls.

* TaxationRequest.ds
* BMLAuthRequest.ds
* CCAuthRequest.ds
* DAVRequest.ds

The following scripts are used to create the objects needed. These are the scripts that customers would change to fit their needs

* CreateCybersourceBillToObject.ds
* CreateCybersourceBMLObject.ds
* CreateCybersourceBMLPaymentCardObject.ds
* CreateCybersourcePaymentCardObject.ds
* CreateCybersourcePurchaseTotalsObject.ds
* CreateCybersourceShipFromObject.ds
* CreateCybersourceShipToObject.ds
* CreateCybersourceTaxationItemsObject.ds
* CreateCybersourceTaxationPurchaseTotalsObject.ds
* CreateCyberSourceTaxRequestObject.ds

These are the pipelines that use the scripts from above and may have need storefront specific integration:

* CybersourceData:CreateBillTo
* CybersourceData:CreateShipTo
* CybersourceData:CreatePaymentCard
* CybersourceData:CreatePurchaseTotals
* CybersourceData: CreateShipFrom
* CybersourceData: CreateTaxItems
* CybersourceData: CreateTaxService

The following is a library script that is used by the Request scripts to build the XML request that is passed to Cybersource. This lib script contain integration independent code and it doesn’t need any storefront specific changes.

* libCybersource.ds

### Templates

Contains templates used by unit test pipeline.

### Configuration Files

Contains two configuration file as follows:

* Cybersource\_PaymentTransaction\_update.xml

Contains custom attributes added to the “PaymentTransaction” object.

* Cybersource-metadata.xml

Contains CyberSource specific site preferences.

# Typical Project Plan

## Roles, Responsibilities

Typically most of the integration works is done by the backend developer. We expect that the person doing this integration is familiar with the web service, xml processing and has hands on experience with the Demandware platform.

## Typical Efforts and Timelines

The level of effort is mostly detected by the services merchant may choose from the CyberSource cartridge. The

|  |  |  |
| --- | --- | --- |
| **CyberSource Service** | **Level of Effort (LOE)** | **Dependencies** |
| Initial Cartridge Setup | **0.5** - Person Day  List of tasks involved:   * Add CyberSource Cartridge to the project * Import Cybersource-metadata.xml * Import Cybersource\_PaymentTransaction\_update.xml | * Cartridge is available |
| Authorize Credit Card | **0.5** - Person Day  List of tasks involved:   * Integrate CyberSource-AuthorizeCreditCard pipeline with COPlaceOrder. | * Merchant ID and Key is established for the client. * Site Preferences for authorization configured with above ID and Key. |
| Address Verification Service (AVS)\* | **0.5** - Person Day | * Initial Cartridge Setup |
| Delivery Address Verification (DAV)\* | **0.5** - Person Day | * Initial Cartridge Setup |
| Bill Me Later (BML) | **0.5** - Person Day | * Setup Account with Bill Me Later. |
| Decision Manager | **0.5** - Person Day | * Access to Decision Manager. * Business rules are defined. * Order status notification URL pointing to Cybersource-NewDecision pipeline is defined. |

\***Note that because customized user interface elements are completely dependent on merchant specification, the time required to interact with the customer to correct address information or confirm standardized address format corrections, is not included; only the time required to integrate with the web services is included, with minimal testing and simplified validation handling, ie. automatically make correction to a customer address, as per validation response.**

# CyberSource Site Preferences

##### Site preference and description

|  |  |
| --- | --- |
| Site Preferences | Description |
| CsMerchantId | CyberSource Merchant ID |
| CsSecurityKey | CyberSource Security Key |
| CsEndpoint | CyberSource Web service End points:  ***Test*** ***https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor***  ***Prod*** [***https://ics2ws.ic3.com/commerce/1.x/transactionProcessor***](https://ics2ws.ic3.com/commerce/1.x/transactionProcessor) |
| CsShipFromCity | Ship to data if fixed for the site |
| CsShipFromStateCode | Ship to data if fixed for the site |
| CsShipFromZipCode | Ship to data if fixed for the site |
| CsShipFromCountryCode | Ship to data if fixed for the site |
| CsBmlMerchantId | BML Merchant ID |
| CsBmlPassword | BML Merchant Key |
| CsBmlItemCategory | BML Item Category |
| CsBmlTCVersion | BML Terms and Condition Version |
| CsBmlNewAcctNo | BML CyberSource account sign-up number |
| CsBmlPromoCode | BML promo code |
| CsBmlPromoMerchantId | BML promo merchant ID |
| CsBmlPromoMerchantPassword | BML promo merchant key |
| CsBmlPromoPromoCode | BML promo promo code |
| CsAvsIgnoreResult | AVS ignore results |
| CsAvsDeclineFlags |  |
| CsDavOnAddressVerificationFailure |  |
| CsDavEnable | This will enable Delivery Address Verification, to help minimize risk of undeliverable or returns orders, because of user data entry errors. |
| CsPaMerchantId | Payer Auth merchant ID |
| CsPaMerchantPassword | Payer Auth Merchant Key |
| CsPaMerchantName | Name |
| CsProductTaxCode | Product Tax Code |
| CsPoaCity | CyberSource purchase order acceptance data – used by Tax |
| CsPoaStateCode | CyberSource purchase order acceptance data – used by Tax |
| CsPoaZipCode | CyberSource purchase order acceptance data – used by Tax |
| CsPoaCountryCode | CyberSource purchase order acceptance data – used by Tax |
| CsPooCity | CyberSource purchase order origin data – used by Tax |
| CsPooStateCode | CyberSource purchase order origin data – used by Tax |
| CsPooZipCode | CyberSource purchase order origin data – used by Tax |
| CsPooCountryCode | CyberSource purchase order origin data – used by Tax |
| CsNexus | CyberSource nexus state list |
| CsNoNexus | CyberSource no nexus state list |
| CsDebugCybersource | To enable/disable debugging |

# Known Issues

<LIST KNOWN ISSUES AND WORKAROUNDS>

# CyberSource document links

1. <http://www.cybersource.com/support_center/implementation/testing_info/simple_order_api/General_testing_info/soapi_general_test.html>
2. <http://www.cybersource.com/support_center/support_documentation/quick_references/view.php?page_id=422>
3. <http://apps.cybersource.com/library/documentation/dev_guides/CC_Svcs_SO_API/Credit_Cards_SO_API.pdf> - Page 163 - Appendix C.
4. <http://apps.cybersource.com/library/documentation/dev_guides/Getting_Started/Getting_Started_Advanced.pdf>
5. <http://www.cybersource.com/support_center/support_documentation/quick_references/>
6. <http://apps.cybersource.com/library/documentation/dev_guides/Payer_Authentication_IG/20090928_Payauth_IG.pdf>
7. <http://apps.cybersource.com/library/documentation/dev_guides/Payer_Authentication_IG/html/>
8. <http://apps.cybersource.com/library/documentation/dev_guides/CC_Svcs_IG_BML_Supplement/html/>
9. <http://apps.cybersource.com/library/documentation/dev_guides/Verification_Svcs_IG/20091012_Verification_IG.pdf>
10. <http://www.cybersource.com/support_center/support_documentation/services_documentation/tax.php>
11. <http://apps.cybersource.com/library/documentation/dev_guides/Tax_IG/Tax_Guide.pdf>

# Release History

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 1.0.0.0.1 | 02/02/2010 | Initial release |