SOP – Braintree Payment Gateway

Prepared For

Iptor IP1

July 2021 –Version 1.0

Table of Contents

Introduction

Braintree Direct is an online payment management gateway system used by businesses worldwide. IP1 customers that use Braintree Direct for their online credit card payment processing can use this document to install Payment Handler and setup and configure Web and IP1 application systems to process payments.

All the installation, setup and configuration including prerequisites and troubleshooting information required for credit card processing using Braintree Payment gateway is included in this SOP document.

It details various Braintree and IP1 configurations and setups and links to Braintree user documentation for detailed information on specific setups.

You can create a test account and test transactions before changing settings to go live with live transactions.

Purpose

The purpose of this document is to assist Iptor consultants' install, setup and configure appropriate business rules in IP1 and Braintree at a customer site for Braintree Payment processing.

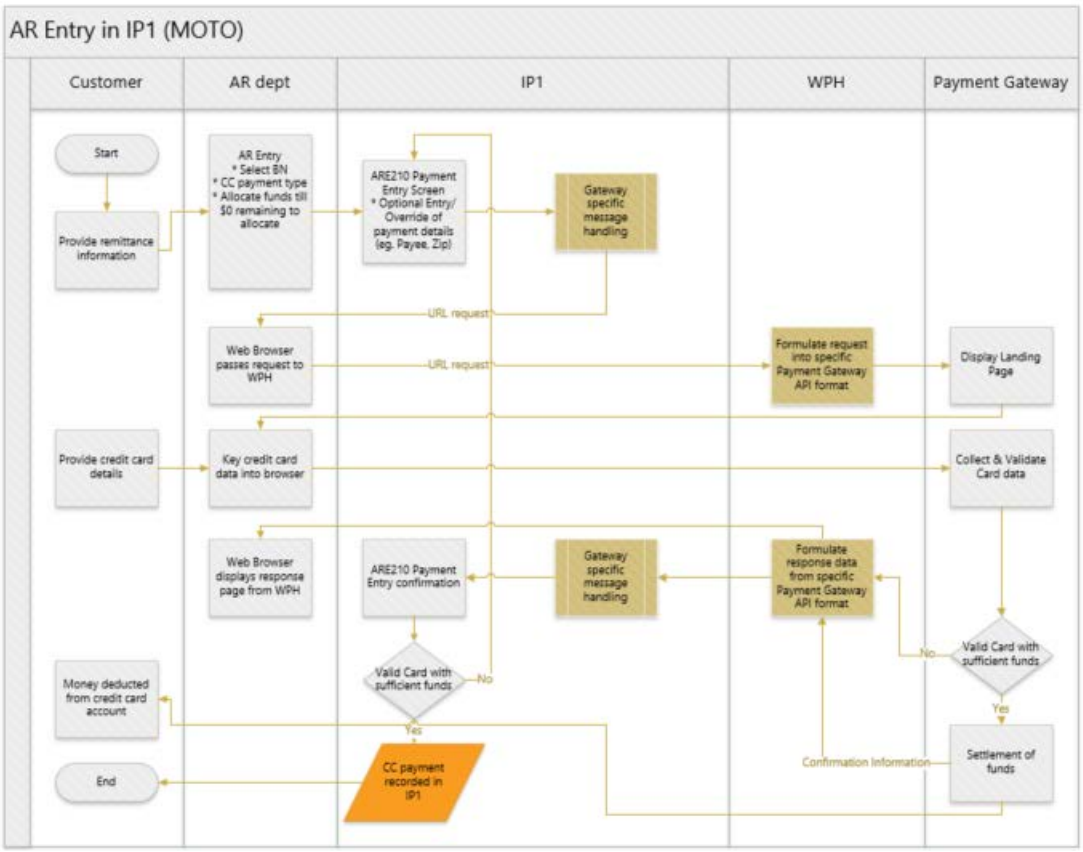
It also includes information the customer will need to assist with provisioning of necessary infrastructure & networking pre-requisites.

Process Flow

AR Entry (MOTO)

MOTO refers to Mail Order / Telephone Order. For us, it refers to cards captured in IP1 AR Entry or Order Entry.

For AR Entry, IP1 integrates with payment gateway landing page for initial capture & validation of card. It requests immediate settlement of full AR payment amount.

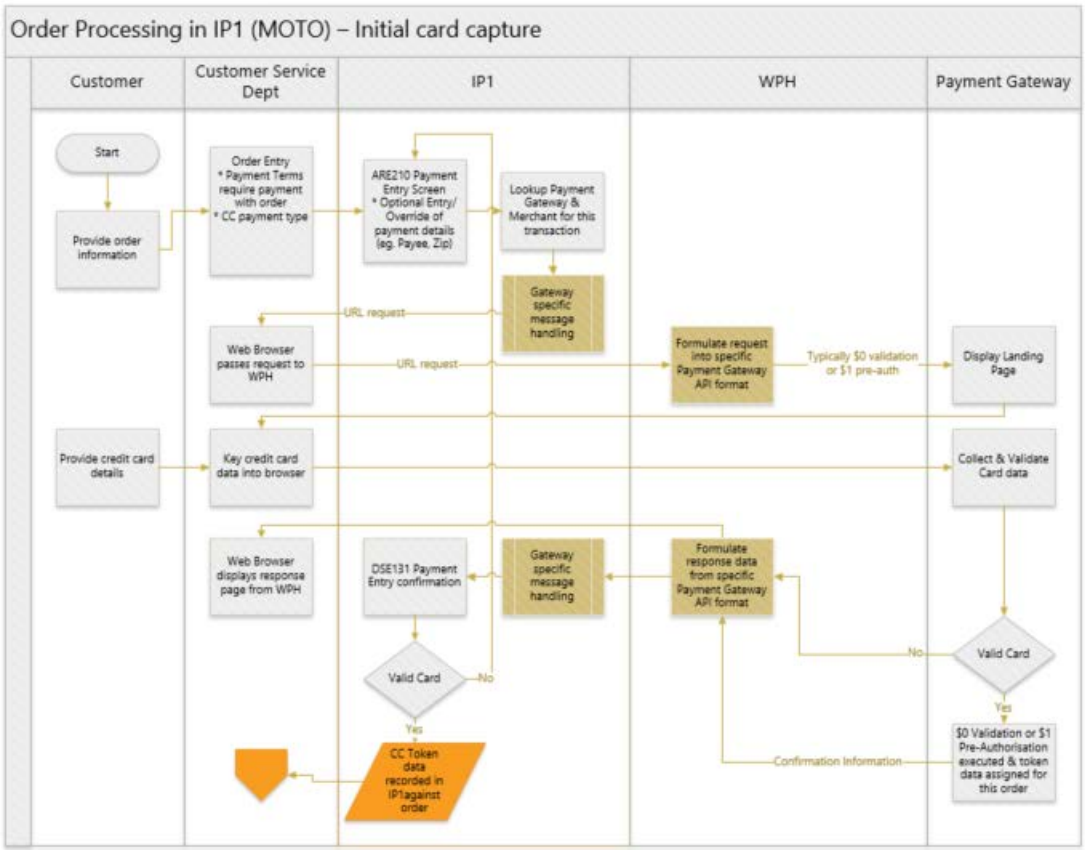

AR Entry in IP1 (MOTO)
Customer
AR dept
IP1
WPH
Payment Gateway
Start
Provide remittance information
AR Entry
* Select BN
* CC payment type
* Allocate funds till $0 remaining to allocate
ARE210 Payment Entry Screen
* Optional Entry/ Override of payment details (eg. Payee, Zip)
Gateway specific message handling
URL request
Web Browser passes request to WPH
URL request
Formulate request into specific Payment Gateway API format
Display Landing Page
Provide credit card details
Key credit card data into browser
Collect & Validate Card data
Web Browser displays response page from WPH
ARE210 Payment Entry confirmation
Gateway specific message handling
Formulate response data from specific Payment Gateway API format
No
Valid Card with sufficient funds
Yes
Valid Card with sufficient funds
No
Money deducted from credit card account
Yes
Confirmation Information
Settlement of funds
End
CC payment recorded in IP1

Order Processing (MOTO)

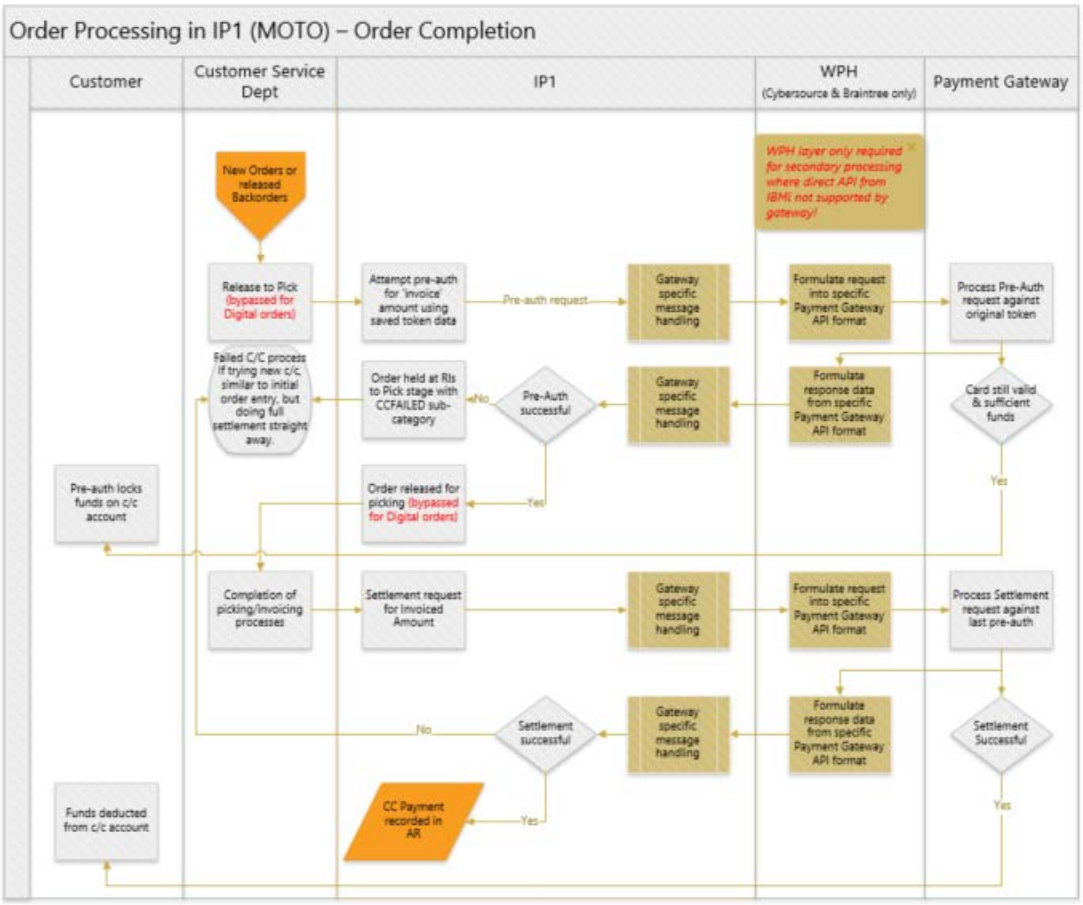
MOTO refers to Mail Order / Telephone Order. For us, it refers to cards captured in IP1 AR Entry or Order Entry.

For orders keyed into IP1 requiring payment with order, IP1 *typically* integrates with payment gateway landing page as follows:

1. IP1 requests $0 or $1 Credit card validation/preauth via payment gateway landing page during order entry stage.
2. Token data for card is stored against order for secondary transaction processing.
3. At release to pick stage, IP1 requests a secondary pre-auth (via gateway's APIs) for the expected invoice amount using the stored token data for this order.
4. If pre-auth is successful, IP1 permits order to be released for picking.
5. Once goods are picked & invoiced, IP1 requests a settlement (via gateway's APIs) against the latest pre-auth for the amount invoiced.
6. If there are any backorders for an order, they repeat steps 3-5 again.
    a. That is, we Settle payments only as goods are supplied, so backorders will issue new pre-auth & settlement requests linked to original token.
7. *Notes:*
    a. *Different configuration options are available to do full pre-auth or charge up front in order entry if preferred.*
    b. *Secondary pre-auth can be configured to occur at Pending status instead of Release to Pick stage.*



Note that secondary transaction processing for some gateways (eg. Datacash or Securepay) do NOT require the WPH layer as those gateways allow direct API integration with IBMi. For Braintree, the WPH layer is required to communicate with Braintree's Server Java SDK APIs, so IP1 calls web service on WPH to perform this.



Braintree fields used by IP1

- Note that we are using the “card tokenisation” features in Braintree to allow multiple pre-authorisations & settlements to occur for a single order.
- During the initial card capture, Braintree will pass information to IP1 including success/failure response details, plus other payment data which IP1 can use for subsequent processing.
- This includes things like masked card number, expiry date, etc
- However, following 2 fields are critical for processing of secondary transactions on such as additional pre-auth, settlements or refunds requests on an order after initial card capture:

Braintree field name	Description / Comment	Max Length	File/Field in IP1 EDI	IP1 field name in XAPCC00P
id	This is the main identifier required for each transaction. This field is **mandatory** to enable IP1 to do any subsequent settlement (or refund in future) on eComm (or MOTO) transactions).	64	DSEDJX00P ‘id’ record	TRNIDE1
paymentMethodToken	This is only required for secondary pre-auth transactions	64	DSEDJX00P ‘paymentMethodToken’ record	CRDID

Infrastructure prerequisites

Communication between servers

This section outlines the communication flow between servers. Note that the arrows (<->) indicate if inbound, outbound or both. The customer will need to ensure that the infrastructure is in place to allow the communication between servers as outlined here.

iSeries -> browser

- When user selects to pay by Braintree c/c, IP1 will 'bounce' to the user's browser, passing a HTML string that will activate the Payment Handler.
- Apart from the Payment Handler URL address, this string includes additional parameters to indicate the IP1 environment, merchant id and merchant account id, transaction type (eg. Sale), default cardholder name from IP1, amount, currency, internal IP1 transaction id,order id (eg. IP1 Cust#). For example…
  https://servername/jph/BTpayment.jsp?clientToken=eyJ2ZXJzaW9uIjoyLCJhdXRob3Jpe&env=TST&merchant=8nxh4th2n9gd66kd&mercacnt=8nxh4th2n9gd66kdIptor&transaction_type=sale&nameOnCard=JBloggs&firstName=JBloggs&lastName=&amount=13.50&currency=EUR&refId=10028040&orderId=E035822&email=norep%40iptor.com&region=L%27&postalCode=L%27Entregu%2FEl+Entrego&countryCodeAlpha2=ES

Payment Handler Web server <-> Braintree server

- The Payment Handler application will formulate the request into Braintree's Client SDK Java API format for entry of credit card details via a secure web page (Braintree Hosted Fields page).
- Braintree will then send a response message back to the Payment Handler application indicating success or failure, but also passing back other payment data which IP1 can then use for subsequent processing.

Payment Handler Web server -> iSeries

- The Payment Handler application will call a program on the iSeries to login & setup the appropriate environment library list.
- It will then call another Payment Manager program in IP1 to send back the response data from Braintree.

iSeries -> Payment Handler web service

- In cases where further 'secondary' credit card transactions are required from IP1, these often occur in the background rather than on an interactive user session.
- For example, Pre-authorisations, Settlements, Refunds on a customer order.
- Rather than IP1 directly calling API on the iSeries to the Braintree gateway, IP1 does a remote call from the iSeries to the Payment Handler Web service to process this transaction utilising previously stored transaction data.
- The Payment Handler will again formulate this into a Braintree Client SDK Java API request and pass back response information to the iSeries as noted above.
- Only difference from initial card capture is that the browser & end user are not involved.

Web Payment Handler Prerequisites

General prerequisites

Following are guidelines on the requirements for the windows server for Web Payment Handler.
Please refer to the appendix for FAQ's on WPH prerequisites.

Windows server

1. Operating system should be Windows Server 2008 R2 or later
2. We recommend 4G RAM minimum as OS needs some.
3. This server may be a Virtual Machine. It does not have to be dedicated physical server.
4. Space on the server required by Web Payment Handler (WPH) will be less than 10MB.
5. Iptor needs a temporary admin account to configure and install the web app.
6. Notepad or Wordpad required for editing configuration files.
7. TLS or SSL Certificate maybe required for Braintree. Refer
   https://developer.paypal.com/braintree/docs/reference/general/best-practices/ruby#tls-certificates
8. This server must have access to the IP1 iSeries environment(s) being used for c/c processing.
9. The server firewall must allow outbound and inbound access to Braintree Payment Gateway.
10. Refer following Braintree URL link to identify specific IP addresses & requirements.
    https://developer.paypal.com/braintree/docs/reference/general/braintree-ip-addresses#braintree-production-ip-addresses

Firewall

Braintree

The Payment Handler Web server will need both inbound & outbound access to the Braintree servers. Following link has a list of the Braintree server IP addresses and URLs which must be allowed.

https://developer.paypal.com/braintree/docs/reference/general/braintree-ip-addresses#braintree-production-ip-addresses

iSeries

The Payment Handler Web server will need outbound access to the iSeries. This is typically done via port 8475.

Install SSL / TLS

Refer

https://developer.paypal.com/braintree/docs/reference/general/best-practices/ruby#tls-certificates

Note: Braintree security requirements could change over time, so it is the client's responsibility to check Braintree support site to ensure they comply with the latest requirements

iSeries

Firewall

WPH Server

The iSeries needs to allow inbound access from the Payment Handler Web server. This is typically done via ports 8472 and 8475.

Java Agent

Each user who enters credit cards in IP1 will need to have the Java Agent installed and running to facilitate the 'bounce' from iSeries to the browser to bring up the Braintree Hosted Fields page.

Braintree products supported by IP1

Following Braintree websites should be referenced to review Braintree products, processes, setup requirements, etc

https://www.braintreepayments.com/dk
https://articles.braintreepayments.com/getstarted/
https://developers.braintreepayments.com/

However, note that IP1 specifically integrates with the **Braintree Direct** product using **Java Client & Server SDKs**. For the checkout UI, we use the **Hosted Fields** solution option rather than Drop-In UK to allow a bit more flexibility & customization if needed.

Following Braintree products & functionalities are NOT currently supported in IP1:

- Braintree Extend product
- Braintree Auth product
- Braintree Marketplace product
- Although Braintree supports a wide variety of payment methods, IP1 supports only for credit cards at present.
- Card Verification features are not currently supported (eg. CVV, Address, 3DS validation)
- Bank Identification Numbers
- Email receipts
- Webhooks
- GraphOI API

Multiple currencies are supported but require set up of separate merchant account id's for each currency.

Setup & Configuration

Business rules for Braintree Direct must be setup with support from Iptor consultants.

Note: The following configurational setups & business rules have to be setup for Braintree Direct. This document does not cover customised setup tasks of specific companies. Deviations from this setup should be covered by setup tasks written by individual companies.

Create Merchant ID(s) & Merchant Account(s)

You will need to contact to setup a Merchant ID & Merchant Account. Braintree will allocate Sandbox (testing) ID & Accounts initially, and you will also require separate production ID & Accounts later.

https://developer.paypal.com/braintree/docs has link to sign up or log in to your sandbox accounts initially.

Note that IP1 can handle links to multiple Merchant ID & Merchant Accounts via the ********/PM-FLD and Merchant Maintenance setups in IP1.

Each Merchant ID will contain at least one Merchant Account, however, if you will be processing multiple currencies, you will need to setup separate Merchant Accounts within the Merchant ID for each currency.

You will need to provide Iptor consultant with the list of Merchant ID(s) and the Merchant Account(s) within those for configuration of the Payment Handler and IP1 control tables.

Team Users & Roles

Once you have login access to the Braintree portal, you will need to add Iptor consultant(s) to list of users, at least for initial Sandbox setup & testing.

1. Login to the Braintree Sandbox portal at https://developer.paypal.com/braintree/docs
2. Click on the 'gear box' icon near top right of screen & select *Teams* option
3. From there you should be able to add new users and decide what roles they have access to

New users should receive a welcome email from Braintree with URL links to relevant documentation to get started.

Public & Private Keys

Each Merchant ID will have a public & private key allocated by Braintree.
These must be configured into Iptor's Payment Handler to allow connection.

1. Login to the Braintree Sandbox portal at https://developer.paypal.com/braintree/docs
2. Click on the 'gear box' icon near top right of screen & select *API* option
3. From there you should be able to generate and view the API Keys (public & private).

Install the Payment Handler

1. You will need 2 installer files from your Iptor consultant:
   1.1. wlp.zip which contains the generic websphere liberty package
   1.2. webapp.zip which contains Iptor's payment handler application
   1.3. These can be placed in a temporary folder for installation purposes, such as c:\temp

2. Extract all C:\temp\wlp.zip file to C:\ which will result in folder C:\wlp
   2.1. You may prefer to install to a different disk drive than C: and that is ok, though it is best to use default \wlp location (default for websphere). Else you will have to do other config amendments.

3. If installing to drive other than C:
   3.1. Go to \wlp\etc folder on that drive and open server.env file in notepad.
   3.2. Change this command to reflect the correct drive name. JAVA_HOME=C:/wlp/java

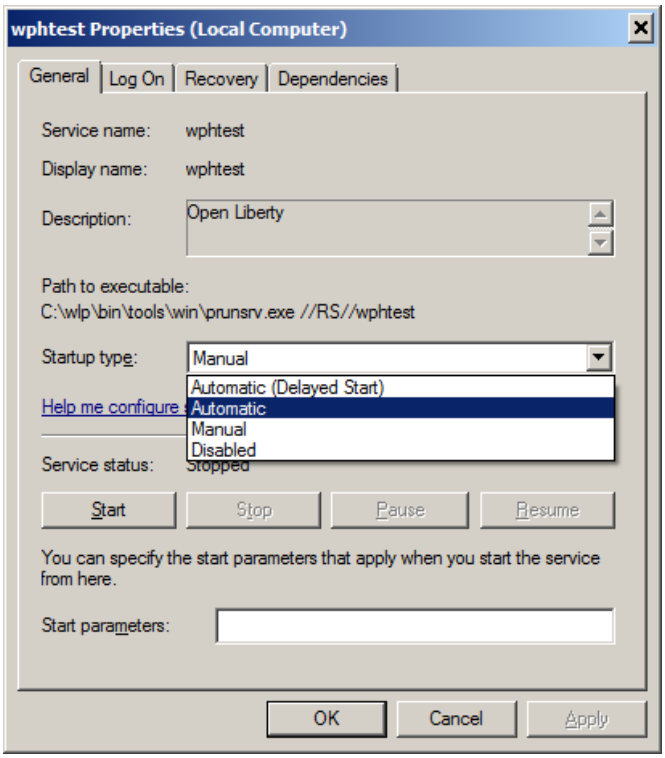
4. Create server(s):
   4.1. **IMPORTANT NOTES**:
      4.1.1. Note: The 'server_name' here will become the name of the service that runs
      4.1.2. If separate web payment handler service instances are going to be run on a single wintel server, then you must assign each one a unique server_name (*wphtest* and *wphlive* for example)
      4.1.3. If completely separate wintel servers are used for live vs QA vs test, then it is probably simplest to use a common server_name for the web payment handler service (*wph* for example)

   4.2. Open command prompt as administrator and change directory to C:\wlp\bin
      4.2.1. Create server - **server create *server_name*** (where server_name is *wph* or *wphtest* for example as per notes above)
      4.2.2. Create Windows service - **server registerWinService *server_name***

   4.3. Using Windows Services configuration tool, set desired running mode to Automatic:

4.3.1. Note: Don't actually start this service till after we've completed installation of Iptor wph app in steps below

4.3.2. You may change the Log On from Local System to another account if desired. This can be a domain or local account, but must have the necessary privileges to read & write files within the /wlp/ folders.



4.4. If running separate instances for test vs live, then repeat steps 3.2 and 3.3 for the additional server_name (*wphlive* for example).

5. Install & configure web application.

5.1. Copy files *server.xml* and *cybs.wsdd* from WebApp.zip into newly created E:\wlp\usr\servers*server_name*\ folder(s), overriding existing files if necessary

5.2. Copy folder jph.war from WebApp.zip to E:\wlp\usr\servers*server_name*\dropins\ folder(s).

5.3. CONTEXT ID – You can skip over this section if you are running a single wph instance with default jph context on this wintel server.

5.3.1. The default installation will use jph.war folder with jph context id settings. This can be left as is if running just one *server_name* instance on this wintel server.

5.3.2. However, if you are running multiple versions of application on one server (eg. wphtest & wphlive), then for the second instance, you must copy folder jph.war to a different name (like jpn.war) in the folder *dropins*, then update file web.xml, section `<display-name>` and file ibm-web-ext.xml, section `<context-root ...>` in the folder C:\wlp\usr\servers*server_name*\dropins\jpn.war\WEB-INF

This will allow to run multiple applications in one server and use different URLs. Eg.

Server_name	**Context**
Wph	jph
Wphlive	jph
Wphtest	jpn

The context id's used here must correlate with ********/PM-FLD URLL entry. For example, we would use following URLL entries for wphlive and wphtest based on example table above.

http://youwebservercomputername/jph/BTGateway
http://youwebservercomputername/jpn/BTGateway

BTree.properties

5.4. Review Braintree system properties file (**btree.properties**). This file is located in each of the E:\wlp\usr\servers*server_name*\dropins\jph.war\WEB-INF\classes folder(s).

5.4.1. Following is example of "btree.properties" file contents. Open it with Notepad, you may see similar as below:

```
# values - test or live
env=test
# allow to override default (jdbc/jph) for the given env.
env.test.jndi.TS1=jdbc/TS1
env.test.jndi.TST=jdbc/TST

# Merchant Id level settings (where mmmmmmmm = merchant id)
env.test.BT_PUBLIC_KEY.mmmmmmmm=???????????
env.test.BT_PRIVATE_KEY.mmmmmmmm=????????????????????????

# Merchant account level (where mmmmmmmm = merchant id, and aaaaaaaa = merchant account)
env.test.USE_CUST.mmmmmmmmaaaaaaaa=true
env.test.USE_ADDR.mmmmmmmmaaaaaaaa=false
```

Notes on BTree.properties structure

- Env
  - env=test Link APIs with Braintree Sandbox
  - env=live Link APIs with Braintree production
- jndi
  - Link to iSeries connection(s) definition in server.xml
  - If this payment handler instance is only ever linked to a single IP1 environment, then you can leave this with the default jdbc/jph setting.
  - If you need to link with multiple environments (eg. MCL, MCT, MCQ, etc), then you should create separate entries for each IP1 environment and corresponding <jndiname> entries in the server.xml file.
  - Also note that TMSWWW/ENV-DFT must be set with correct IP1 env id corresponding to these.
- BT_PUBLIC_KEY and BT_PRIVATE_KEY are defined with merchant ID (MERCHANT in PM-FLD)
  - These must the public & private keys for the merchant id. The details can be found below
  - Locate your sandbox API credentials
- Following settings are defined with both merchant ID and merchant account ID (MERCACNT in PM-FLD):
  - USE_CUST turns on and off setting customerID to our TRNID
    - Mandatory parameter. Valid settings are 'true' or 'false'
    - Set as true to facilitate secondary pre-auth processing
  - USE_ADDR turns on and off passing IP1 address to BT
    - Mandatory parameter. Valid settings are 'true' or 'false'
    - *Set as true to turn on address validation.*

- *Note that as at 24/6/21, this is not really used, as we don't use the 'Card Verify' processing*

- PAGE_SFX defines html name for given BT merchant account ID.
  - Optional parameter.
  - If left out, will default to using following landing page: BTpayment.jsp
  - If given, this parameter identifies suffix for the file name. eg.

```
env.test.PAGE_SFX.mmmmmmmmaaaaaaaa =US
```

    - This setting would use file name BTpaymentUS.jsp
  - It allows different page look for different merchant accounts (eg. Diff language or logo)
  - Refer 'Appendix: Customizing the Card Capture screen (BTpayment*.jsp) for more information

- Other notes:
  - Braintree does not require any URL setting to point to relevant test or production server (unlike DC.properties which has DCurl setting).
  - Braintree is 'black box' and uses the ***test*** or ***live*** identifier to choose sandbox or production processing.

Server.xml

5.5. Review server.xml file(s) to connect with iSeries environments. This file is located in each of the C:\wlp\usr\servers*server_name*\ folder(s).

5.5.1. Ports for listening for requests from the iSeries:

5.5.1.1. By default, the *server.xml* contains following code which defines the listening ports that will be used for HTTP or HTTPS. It defaults to 80 and 443 respectively.

```
<httpEndpoint host="*" httpPort="80" httpsPort="443" id="defaultHttpEndpoint">
```

5.5.1.2. The ********/PM-FLD URLL settings in IP1 will typically use http, but you may prefer to use https. Obviously, this will impact which of the 2 port settings above are relevant.

5.5.1.3. Typically you will either use 80 or 9080 for http and 443 or 9443 for https

5.5.1.4. You may need to change the default port setting in *server.xml* if those default ports are already used by another application, or if you have multiple *server_name* instances of payment handler running on same wintel server (eg live vs test).

5.5.1.5. Use windows command prompt and netstat command to check if ports are being used. Eg.

5.5.1.5.1. netstat -aon | findstr 80

5.5.1.5.2. netstat -aon | findstr 443

5.5.1.6. For example, if you already have a live instance using ports 80 / 443, you could set the test instance to use 9080 / 9443 as follows:

```
<httpEndpoint host="*" httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint">
```

5.5.1.7. Note: If migrating from old .Net version of Web Payment Handler on same server, then you will likely need to end the old .net website in IIS, then repeat step 5.5.2.5 to check ports are available.

5.5.1.7.1. Go into IIS

5.5.1.7.2. Click on the website(s) for wph (eg. Phlive, phtest)

5.5.1.7.3. On right hand side, click on the option to stop that service.

5.5.2. iSeries environment connection(s) settings

5.5.2.1. Within the *server.xml*, look for iSeries connection properties (eg. Search for jndiName). For example, you will find something like below.

```
<dataSource beginTranForResultSetScrollingAPIs="true" id="DefaultDataSource"
isolationLevel="TRANSACTION_READ_UNCOMMITTED" jdbcDriverRef="jtDrv"
jndiName="jdbc/jph" type="javax.sql.DataSource">
```

```
<properties.db2.i.toolbox dateFormat="iso" naming="system" password="TMS"
serverName="IBSBKDEV" timeFormat="iso" user="TMS"/>
</dataSource>
```

5.5.2.2. If this payment handler instance is only ever linked to a single IP1 environment, then you can leave this with the default jdbc/jph setting (must match jndi setting in btree.properties file).

1.1.1.1. If you need to link with multiple environments (eg. TS1, TS2, TS3, etc), then you must create separate dataSource entries for each IP1 environment, where each <id> and <jndiname> corresponds with the <jndi> settings in btree.properties file. For example, replace jph default with following for TS1 env:

```
<dataSource beginTranForResultSetScrollingAPIs="true" id="TS1"
isolationLevel="TRANSACTION_READ_UNCOMMITTED" jdbcDriverRef="jtDrv"
jndiName="jdbc/TS1" type="javax.sql.DataSource">
<properties.db2.i.toolbox dateFormat="iso" naming="system" password="TMS"
serverName="IBSBKDEV" timeFormat="iso" user="TMS"/>
</dataSource>
```

5.5.2.3. The user and password entries must correlate with a valid iSeries userid (refer separate section later on iSeries configuration).

5.5.2.4. If the iSeries user has default jobd/library list that includes the relevant object library containing XAS010 and XAS000* service programs, plus the relevant data library for this IP1 environment, then no additional configuration

5.5.2.5. If the iSeries user is not setup with library list including the appropriate IP1 environment data library and object library(s), then you will either need to revise that user to setup jobd with appropriate libl, or you can include additional libraries statement in the dataSource as per example below:

```
<dataSource beginTranForResultSetScrollingAPIs="true" id="DefaultDataSource"
isolationLevel="TRANSACTION_READ_UNCOMMITTED" jdbcDriverRef="jtDrv"
jndiName="jdbc/jph" type="javax.sql.DataSource">
<properties.db2.i.toolbox dateFormat="iso" naming="system"
password="fr0gha1r$" serverName="IBSBKDEV" timeFormat="iso" user="TMS"
libraries="TMSBASE,TMSDTATS1,TMSOBJZX1"/>
</dataSource>
```

5.5.2.5.1. Note that the libl must include:

5.5.2.5.1.1. TMSBASE library containing SETUP command

5.5.2.5.1.2. Data library for this IP1 environment

5.5.2.5.1.3. Object library(s) containing XAS010 and XAS000* objects.

5.6. If you wish to customize the web payment handler card capture page, refer Appendix: - Customizing Card Capture page (BTpayment*.jsp)

5.7. If you wish to customize the web payment handler receipt & response pages, refer Appendix: Customizing Receipt & Response web page (BTresponse.jsp)

5.7.1. For MHE, this was already done during the original .Net implementation

6. For each server_name, you can now go and start the service. Follow instructions in section 'Restarting the Server'. In MHE case, just wph service to be started.

7. Configuration network connection

7.1. Ports listed in files server.xml should be opened and https connectivity configured to the relevant payment system.

7.2. SQL ports should be opened to iSeries (default ports are – 449,8470,8471,8475,8476).

7.3. See section 'iSeries configuration' for the user profile and SQL stored procedure requirements.

Restart the server

Instructions to restart services:

- There two ways to do it as outlined below.
- For either method, replace *server_name* as appropriate (eg. wph, wphtest, wphlive)

Method 1/

- Stop relevant Windows service with this *server_name*
- Check the log file (C:\wlp\usr\servers*server_name*\logs\messages.log) to make sure server has stopped and then start Windows service again.
- Unfortunately, Windows services option 'restart' can hang Java process quite badly, therefore it is necessary to do separate stop and start.

Method 2/

- Open command prompt as Windows administrator.
- CD c:\wlp\bin (or relevant folder where Open Liberty bin folder is).
- Use the following command to stop the server: **server stopWinService server_name**
- Check the log file (C:\wlp\usr\servers*server_name*\logs\messages.log) to make sure server has stopped and then start Windows service again.
- Use the following command to start the server: **server startWinService server_name**

iSeries configuration

User profile

The Payment Handler application needs a dedicated iSeries user profile to be created to support communication of Braintree response data back to the iSeries. The profile would only be used by the Payment Handler web server to log on, connect to correct environment and call programs. It should ideally be only used for this machine login, and not for human login.

Key requirements for this login are:

1. Password is *NONE . . . . . : *NO
2. Password expiration interval . . . . . : *NOMAX
3. User class . . . . . : *PGMR
4. Special authority . . . . . : *IOSYSCFG
   *SPLCTL
   a. Ok, but not necessary for other authorities to be given also (e.g. *ALLOBJ)
5. Group profile . . . . . : QPGMR
6. Owner . . . . . : *GRPPRF
7. Current library . . . . . : TMSBASE or IBSBASE
   a. Depending on IP1 implementation
   b. This library includes the SETUP command which is used to library list for the IP1 environment. If customer has their SETUP command included in an alternative library, then that library name must be used.
8. Job Description / Library List
   a. Ideally, this user profile should include full library list for the relevant IP1 environment
   b. In cases where the web payment handler wintel server may need to talk to multiple IP1 environments, consider creating separate iSeries user ids for each environment with correct library list. This can be achieved by linking to an appropriate JOBD for that env.
   c. Alternatively, the server.xml configuration on wintel server can include libraries statement to identify the object & data libraries.
   d. Critical requirement is:

- The base object library containing XAX000N e.g. TMSOBJZX1
- The object library containing XAS010 if that is elsewhere.
- The data library containing XACTD00P e.g. TMSDTATS1

XAX000 Stored Procedure
On the iSeries, create stored procedure using the following SQL statement:
This should be created in the primary data library for each environment used by web payment handler.
For customers already using IP1 web services, this should already be there!10

```
CREATE PROCEDURE TMSDTAxxx/XAX000 (IN ENV CHAR(3),
IN JBUS CHAR(10),
IN JBID CHAR(10),
IN PGMN CHAR(10),
OUT STS CHAR(1),
IN XIN blob(2M),
OUT XOUT Blob(16M),
OUT XMSG Blob(64K))
LANGUAGE RPGLE
PARAMETER STYLE GENERAL
NOT DETERMINISTIC
DYNAMIC RESULT SETS 0
EXTERNAL NAME TMSOBJZX1/XAX000N
```

Names of the libraries (in red) should correspond to ones in your environment.
Make sure that stored procedure is unique.

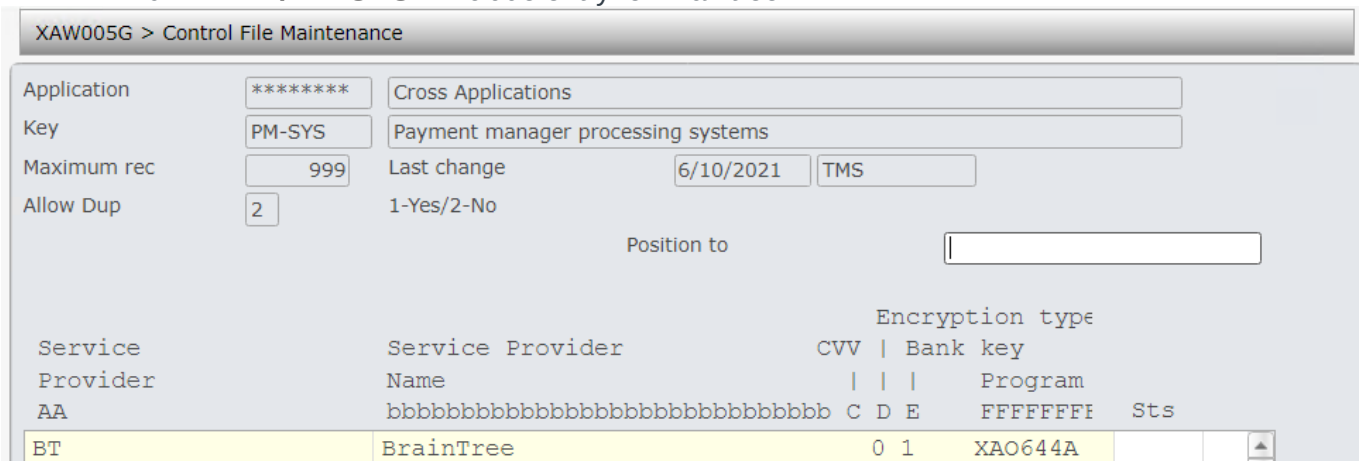
IP1 Configuration

Change the web address of IP1 web invoke and web service call to the associated web location that is deployed.
If you installed SSL, make sure you use https from IP1.

1. Install Java Agent. This is required to open the Hosted Fields page for credit card entry
2. Deploy RPG program XAO644A if not already deployed.
    a. Already there for MHE (using with .Net version already)

3. In IP1 create/update control file entries for the following control files:
    a. TMSWWW/ENV-DFT
        i. For each IP1 environment, this control file must contain the matching environment id.
        ii. This is used by the web payment handler to identify which jndi settings to use in btree.properties file.
        iii. This is particularly critical where single instance of web payment handler can talk to multiple IP1 environments!

    b. *********/PM-SYS** – include entry for Braintree:



c. **********/PM-FLD – setup Braintree entries**

Critical field settings:

- MERCHANT is the BrainTree merchant ID
- MERCAACNT is the BrainTree merchant account ID
- URLL – The URL used for initial card capture via browser bounce to the payment handler. Typically: http://yourwebservercomputername/jph/BTGateway
- URL and URL1 – The URL used for secondary transaction remote calls to the payment handler service. http://yourwebservercomputername/jph/services/CSServiceSoap
- PN = 1 (pass Process Number rather than Customer Number for additional Custom Data in Braintree reporting portal on secondary transactions)

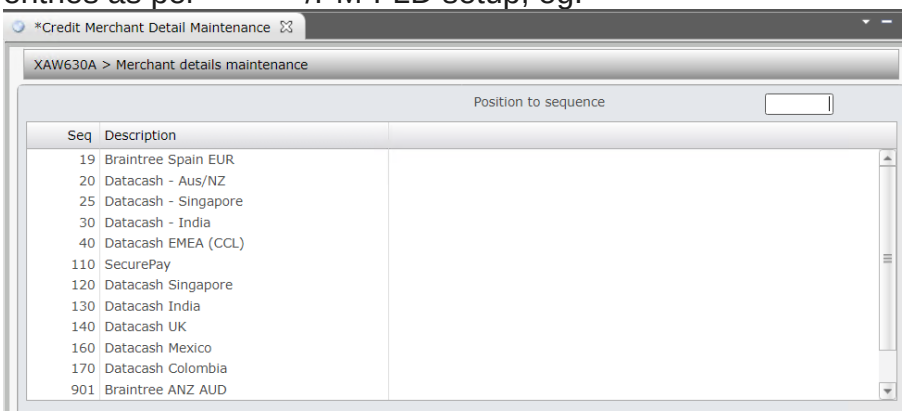
Generic Iptor sandbox merchant example:

```
Service Provider
|  Merchant Id
|  |         Field    Value
AA BBBBBBBB cccccccc DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
BT TEST     DEBUG    1
BT TEST     EMAIL    norep@ibs.net
BT TEST     MERCACNT Iptor
BT TEST     MERCHANT 8nxh4th2n9gd66kd
BT TEST     PAD1     Address1
BT TEST     PAD2     Address2
BT TEST     PAD3     City
BT TEST     PAD4     State
BT TEST     PAD5     Post_Code
BT TEST     PAD6     Country
BT TEST     PN       1
BT TEST     URL      http://servername/jph/services/CSServicesSoap
BT TEST     URLL     http://servername/jph/BTGateway
BT TEST     VLDAMT   0100
BT TEST     WAIT     060
BT TEST     WRNCDE   200201230520
```

d. **********/PM-OPT**:

i. Nothing required here for Braintree

4. Configure **Merchant Detail Maintenance** (XAW630A) to use relevant Braintree account entries as per ********/PM-FLD setup, eg.



5. Increase 'XA-TRNID Credit card transaction ID' control number to higher round number (for example - '00100000').

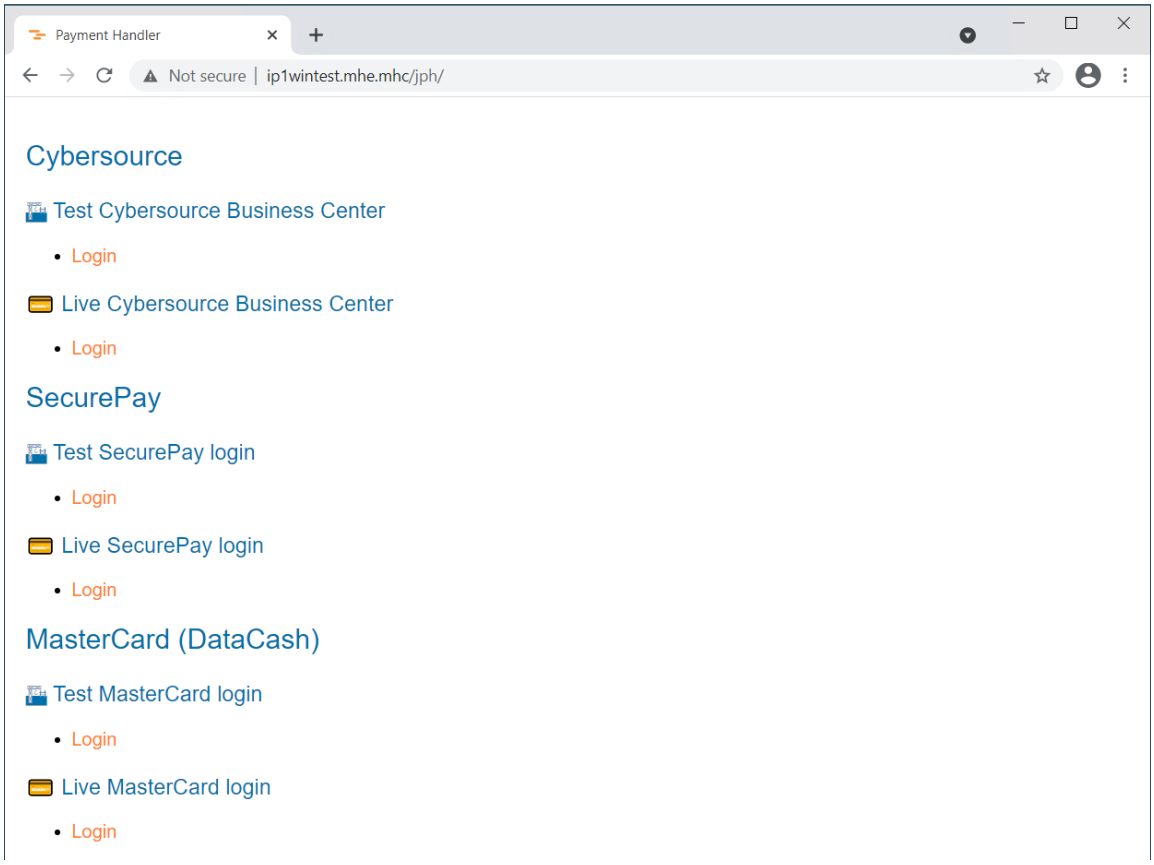
Testing

Check WEB Payment Handler Install

1. Test ability to connect to Open Liberty
    a. After starting the service, on browser (from own PC if you are in same network or VPN as the payment handler wintel server), type URL to check if it gets a response.
        http://dnsservername:port/ or http://ip:port/
        i. Port number can be left off if using default http port 80
    b. This verifies that user can connect over the network/VPN to Open Liberty using the designated port.
    c. On successful connection you should see below:

    d. If you get an error, then refer to Troubleshooting section below.

2. Test ability to connect to the wph application
    a. Following above test, could include the /jph portion of URLL to also check that the wph application itself is running ok.
        http://dnsservername:port/jph/ or http://ip:port/jph
    e. On successful connection you should see screenshot below:
        i. Note that this is a static page in the wph application. Presence of this page simply proves the wph application is running ok.



3. HTTPS testing
    a. If you wish to use https instead of http for the connection from user's browser to the wph, then they will first need to install & setup certificate in Open Liberty.
    b. Refer following link for information on how to set this up. https://www.ibm.com/support/knowledgecenter/en/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/twlp_sec_comm.html
    c. Then repeat test 1 & 2 above with https instead of http prefix.

Check iSeries link to/from wph

1. Setup Java Agent to link to relevant iSeries.
    a. If Java agent doesn't connect:
        i. On iSeries, type command NETSTAT *CNN
        ii. Look for an entry that shows port 4454 listening.
        iii. If not found, CALL XAO255C to start agent listener on iSeries (sign off & on before doing this command to clear libl. i.e. Don't want IP1 environment specific libl).
        iv. Then recheck NETSTAT *CNN to confirm port 4454 is now showing ok
        v. Retry starting the agent on PC.
2. On the iSeries
    a. Update ********/PM-FLD control file to new URL pathnames
    b. Do initial verification testing using XAO630T
        i. Notes:
            1. Can do this test with btree.properties unaltered from iptor_tst defaults (i.e. Using Iptor test merchant and keys). Should be able to get a valid acceptance response on card 41111111111111111
            2. This test verifies:

a. All being well, should bring up the Braintree landing page, accept test card info, and pass valid response back to the iSeries.
b. If not, could imply one of following isn't correct.
c. ********/FLD URL* settings
d. Java agent setup on individual's PC or on iSeries
e. WPH software installation & config
f. Ports/firewall connections between WPH server & Braintree, and between WPH and iSeries.

3. You may need to create a dummy first sequence entry in Merchant Maintenance with catch all wildcard asterisks in criteria fields to force use of particular test merchant account since XAO630T doesn't know the company, payment type, etc.
4. XAO630T currently uses hard coded AUD currency, so this may cause failures with some overseas merchant ids.

ii. CALL XAO630T
iii. Key request TC or CC using default card and reference information.
iv. E.g. Card 4111111111111111, CVV 123, Expiry date after today's date. Amount=$10.08, Our Ref=date/timestamp value
v. Press Enter
vi. Should bring up landing page for your Braintree payment gateway.
vii. Key card information again.
viii. Check appropriate response comes back to the iSeries program.

End to end testing with IP1

1. Ensure appropriate AR Payment types and other configuration preferences are setup.
2. Ensure Merchant Maintenance links to appropriate test or live merchant setups in ********/PM-FLD
3. AR Entry test
    a. Key payment using appropriate AR Payment Type(s) for c/c link with relevant payment manager.
    b. After finalising allocation of amount, should bounce to relevant payment manager host page for entry of card details to pay matching amount requested on the IP1 screen.
    c. Response should be returned to ARE005 screen.
4. Order Entry tests
    a. Key order with prompt payment terms and select payment with appropriate AR Payment Type(s) for c/c link with relevant payment manager.
    b. Should bounce to relevant payment manager host page for entry of card details. Amount and method (preauthorization vs immediate payment) will depend on configurations. Refer SOP documentation.

Troubleshooting

If URL brings up an IIS screen, this indicates that the designated port is already allocated to an IIS website.

a. You may need to change configurations to use a different port, or stop the website in IIS

If URL brings up error 404 – File or directory not found error on browser.

b. Check all installation steps have been completed on both windows and iSeries.
c. Check that the relevant windows service is running.
d. You may need to include specific port# in the ********/PM URL* field settings, rather than relying on default.
e. If still issues, check log files in C:\wlp\usr\servers*server_name*\logs
    i. Messages log file has I, A and E entries. Look for E (error) messages.
    ii. For example, following error line indicates that the port configured in server.xml file could not be allocated.
    [1/5/21, 18:45:44:454 EST] 00000026 com.ibm.ws.tcpchannel.internal.TCPPort
    E CWWKO0221E: TCP Channel defaultHttpEndpoint initialization did not succeed. The socket bind did not succeed for host * and port 80. The port might already be in use. Exception Message: Address already in use: bind
f. For port usage clash,
    i. Refer to the following for instructions to check which application(s) are using a port https://www.printsupportcenter.com/hc/en-us/articles/115003386949-Determine-which-program-uses-or-blocks-a-port
    ii. If clash exists, you may need to end the other applications using this port, or change server.xml and PM-FLD URL* to use a different port.

Unsuccessful response from payment gateway

a. It is quite possible that the Receipt screen may show 'valid' errors in the normal course of entry (e.g. Expired card, invalid CVV, etc.). However, there may be other cases where problem exists in programming or setup.

Top left corner of response page shows ""&#x1F3D7 Receipt". What does this mean?

a. This is Unicode character "Building Construction" (https://www.compart.com/en/unicode/U+1F3D7) showing we are in test mode
b. It should show up as following image🏗 , however Internet Explorer can't show it.
c. Other browsers display it ok, so try changing your default browser, or just ignore this.

Other checks – IP1 log files

a. You may need to check & verify URL being called and the response received.
b. Check XAPCA, XAPCB, XAPC files for payment handler.
c. Check Braintree, log file - XAPCD00P.

Braintree Direct – Go Live

Once you are satisfied with the test, you can contact Braintree to arrange go live service and get access and authorization to go live. Before going live ensure the configuration is updated for live transactions as outlined under Web Application Configuration and IP1 Configuration.

Appendix

Appendix: FAQ – Web Payment Handler (WPH) prerequisites

a) Do we need separate installation & configuration for each gateway or merchant id? (e.g. Braintree APAC live, Braintree APAC test, Braintree EMEA merchant for Live, Braintree EMEA merchant for test, Datacash APAC live, Datacash APAC test)?
Single or separate installation is supported. Currently Iptor have only one deployment file which can be used for all supported Payment Gateway merchants. Each gateway will require its own customised configuration file specifying client preferences and merchant(s) information.

b) Even though there is one deployment file, do we need to do multiple installs to handle the different merchant Ids, or can we handle multiple configurations/merchants within single installation?
This is up to client to decide. A single WPH server can handle multiple URLs, so it is possible to do one installation of the software but set up separate URLs for each Merchant account. Each URL would be linked to a separate copy of the WPH configuration file which is specific to that merchant account. This also means we can allow multiple iSeries environments with one deployment of the WPH software, as the configuration file for each URL will point back to the appropriate iSeries environment.

Generally, Iptor suggest that it is simpler to have a single WPH server and web payment handler software install, with different URLs & configuration files for test vs live or other separate merchant accounts.

However, clients may prefer to install multiple copies of the application onto one server, or onto multiple servers.

c) Are there any requirements for number of ports?
No specific requirements, default, HTTP is still using port 80, and SSL port uses 443. Ports can be customised by client's own IT department, and SSL can be added and specified by the clients also.

d) Would there be any problems if installed within same VM as other applications (e.g. Book Production file server), or do we suggest better to keep this WPH in separate VM?
Our landing page can be with any other applications, as it only consumes very little resource, and it has very high security features.

Appendix: Customizing the Card Capture screen (BTpayment*.jsp)

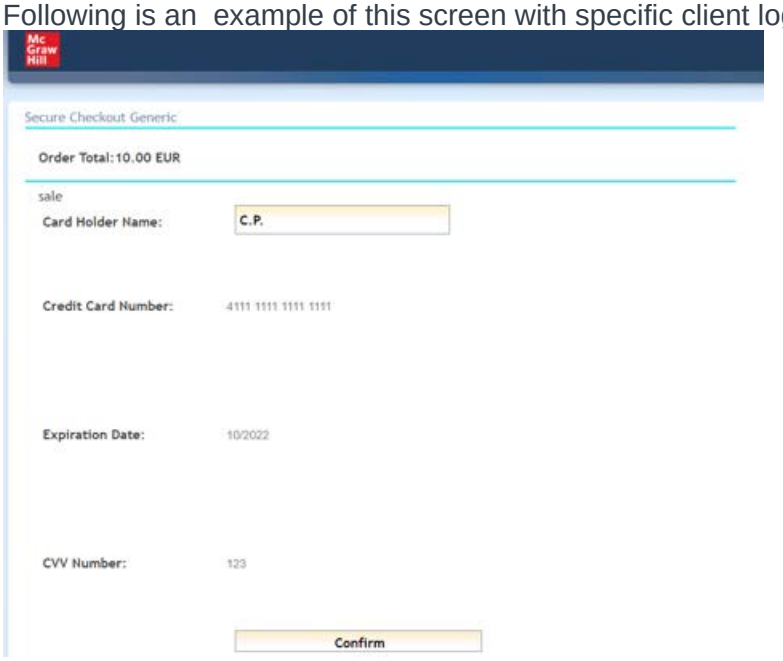
Since IP1 is using the Hosted Fields UI option, we have a small amount of flexibility to customize the card capture entry screen.

- Iptor provide a default BTpayment.jsp file which can be used.
- This is located in E:\wlp\usr\servers\wph\dropins\jph.war
- Customers can optionally copy & create their own customized vesions of BTpayment.jsp for define different Merchant ID / Account ID combinations if desired.
- For example, you may wish to create a copy for different language instead of English for the field labels, or to perhaps use a different logo.
- If creating customized copy(s), give each version a unique suffix.
    - Eg. Could use BTpaymentSP.jsp for Spanish version.
- You would then need to edit the PAGE_SFX settings in Btree.properties to link relevant merchant accounts with relevant suffix id for this file.

Important Note:

1. If you wish to include new fields or remove any fields, please talk to your Iptor consultant first, as this may enhancements & customization of our Payment Handler software also.
2. For example, we have specific coding in place for following fields.
    a. Transaction Type
    b. Amount, Currency
    c. Default cardholder name

Following is an example of this screen with specific client logo:



Appendix: Customizing Response (Receipt) web page (BTresponse.jsp)

1. Receipt / Response page -BTresponse.jsp
   1.1. This page will show response data from the payment gateway.
   1.2. If user is not interested in some of this information, they can simply close the browser tab.
   1.3. Unfortunately, we cannot completely prevent this screen showing or force auto close by the browser, however, you can edit the contents shown by editing the JSP.
   1.4. Following is an example of response from Braintree for a successful payment:

   1.5. Customizing to hide selected fields.
      1.5.1. Find the BTresponse.JSP file in E:\wlp\usr\servers\wph\dropins\jph.war
      1.5.2. Edit the BTresponse.JSP file with preferred editor.
      1.5.3. For example, you may wish to hide the payment token field.
      1.5.4. You can comment (use /* */) or just delete the lines altogether from the BTresponse.jsp

```
if (token != null)
    out.println("payment token   : " + token);
```

BTresponse.jsp

```
31      decision     = request.getAttribute("status").toString();
32  if (request.getAttribute("processorResponseCode") != null)
33      reasonCode   = request.getAttribute("processorResponseCode").toString();
34  if (request.getAttribute("processorResponseText") != null)
35      message      = request.getAttribute("processorResponseText").toString();
36  if (request.getAttribute("id") != null)
37      refNumber    = request.getAttribute("id").toString();
38  if (request.getParameter("refId") != null)
39      reconcileId  = request.getParameter("refId").toString();
40  if (request.getAttribute("token") != null)
41      token        = request.getAttribute("token").toString();
42  if (decision != null)
43      out.println("decision        : " + decision);
44  if (reasonCode != null)
45      out.println("reasonCode      : " + reasonCode);
46  if (message != null)
47      out.println("message         : " + message);
48  if (refNumber != null)
49      out.println("ref number      : " + refNumber);
50  if (reconcileId != null)
51      out.println("trn. Id         : " + reconcileId);
52  if (token != null)
53      out.println("payment token   : " + token);
54
55      List<String> keys = Collections.list(request.getParameterNames());
56      Collections.sort(keys);
57      List<String> keysAttr = Collections.list(request.getAttributeNames());
58      Collections.sort(keysAttr);
59  %>
```

For unsuccessful response refer to the Troubleshooting section

Other Tips

Installation summary if migrating from old .Net to new Java version of wph

1. Note that Braintree is only supported on the new Java version of wph!
2. Wintel setup
    a. As per Install the Payment Handler section in this document
3. iSeries
    a. You may already have a common user profile that you were using with the .Net version which can be used again here.
    b. Review if you need to amend user profile jobd/libl, or alternatively set the libraries keyword in the se in the server.xml
4. IP1 config
    a. New XAX000 stored procedure must be created in each IP1 environment.
    b. TMSWWW / ENV-DFT – must be set to matching IP1 environment code for each IP1 environment.
    c. ******** / PM-FLD
        i. Change all URLL entries for Braintree service provider
            1. Use /jph instead of /phlive or /phtest
            2. Other suffixes may be required if you used multiple Context Ids during the Payment Handler Install and configuration.
        ii. Eg. Migrating from Datacash on .net WPH to Braintree on new Java WPH version
        From
        http://dnsservername /ph/DCGateway
        to
        http://dnsservername /jph/BTGateway

IP1 Test environment refresh from live (and using live vs test merchant/cards)

1. TMSWWW / ENV-DFT must be reset to relevant environment id
2. ******** / PM-FLD
    a. Suggest that it is best practice to include both LIVE and TEST merchant entries here, so that you can simply use Merchant Details Maintenance to point to relevant one for this environment.
3. To point IP1 environment to TEST instead of LIVE wintel & merchants…
    a. Go into menu opt 50,60. Credit Merchant Detail Maintenance
    b. Review each of the sequence entries for Datacash provider and swap the merchant id from LIVE to TEST.
    c. If you have multiple merchants, you may have something like xxTEST or xxLIVE merchants setup in ******** / PM-FLD
    d. Restart IP1 background processing jobs to ensure changes picked up
4. Of course, if you want to point test environment to live merchant/card, then simply revert the setups in step 3

Creation of new IP1 environment

1. If you create a new IP1 environment that you want to handle Datacash card payments, then you will need to:
2. Wintel server
    a. Add new IP1 environment entries into the btree.properties and server.xml files
    b. If this is a test IP1 environment, then suggest that you do this on both dev and prod wintel servers, so that you can point the IP1 environment to both test and live merchants.
    c. Restart wph services
3. IP1 config
    a. Ensure stored procedure XAX000 is created in this environment

b. As noted on IP1 Test environment refresh from Live

Object library change (e.g. Upgrade to IP1)

1. If new versions of XAX000 or XAS010 service programs are created, then you will need to do following:
2. iSeries userid
    a. Check if jobd/libl needs updating
3. Wintel servers
    a. If server.xml <datasource> tags are currently using libraries="TMSBASE,TMSDTAxxx,TMSOBJZ78"
    b. You will need to change each <datasource> tag to the new object library (eg. TMSOBJZ78 may change to TMSOBJZXI).
    c. Restart the wph services
4. IP1
    a. Delete and recreate the XAX000 stored procedure, pointing the EXTERNAL NAME setting to appropriate library containing XAX000N object

```
EXTERNAL NAME TMSOBJZXI/XAX000N
```

Linking customized Card Capture pages with merchant account

1. Refer Appendix: Customizing the Card Capture screen (BTpayment*.jsp)

New Merchant ID or Merchant Account

1. To set up a new merchant id or merchant account…
2. Identify the Merchant ID, Merchant Account, Public & Private keys.
    a. Refer Create Merchant ID(s) & Merchant Account(s)
    b. And Public & Private Keys
3. Setup ******** / PM-FLD merchant entries
    a. Copy control file entries from one of the existing Braintree merchant id's and change following settings to match the new merchant details:
        i. MERCHANT is the BrainTree merchant ID
        ii. MERCAACNT is the BrainTree merchant account ID
4. Wintel servers
    a. btree.properties
        i. Refer BTree.properties section of this document for details.
        ii. Note that BT_PUBLIC_KEY and BT_PRIVATE_KEY are set at merchant id level, while other settings are at merchant id / merchant account level.
        iii. Following is a generic example:

```
# Merchant Id level settings (where mmmmmmmm = merchant id)
env.test.BT_PUBLIC_KEY.mmmmmmmm=???????????
env.test.BT_PRIVATE_KEY.mmmmmmmm=????????????????????????

# Merchant account level settings (where mmmmmmmm = merchant id, and
aaaaaaaa = merchant account)
env.test.USE_CUST.mmmmmmmmaaaaaaaa=true
env.test.USE_ADDR.mmmmmmmmaaaaaaaa=false
```

    b. After editing the btree.properties, you will need to restart the service. Refer Restart the server