SOP – SecurePay Payment Gateway

Prepared For

Iptor IP1

March 2021 Version 2.1

Table of Contents

Introduction

SecurePay is an online payment management gateway system used by businesses worldwide. IP1 customers that use SecurePay 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 SecurePay Payment gateway is included in this SOP document.

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' setup and configure appropriate business rules in IP1 and SecurePay at customer's site for SecurePay Payment processing using the landing page PCI Compliance functionality. This landing page functionality is required for credit card payments for back end IP1 i.e. for Payment with Order Entry, AR Receipts etc.

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

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 SecurePay c/c, IP1 will ‘bounce’ to the user’s browser, passing a HTML string that will activate the Payment Handler.
- Aside from the Payment Handler URL address, this string includes additional parameters to indicate the IP1 environment, type of transaction (pre-auth vs immediate settlement), amount, currency, merchant reference number, merchant ID, and the IP1 customer or process number.

Payment Handler Web server <-> SecurePay server

- The Payment Handler application will formulate the request into an XML request to activate SecurePay Hosted Pages Solution (HPS) for entry of credit card details via a secure web page.
- SecurePay will then send a response message back to the Payment Handler application indicating success or failure, but also including token & reference details which can be used for subsequent processing.
- Note that we are using the Card Tokenisation features in SecurePay to allow multiple pre-authorisations & settlements to occur for a single order.

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 SecurePay.

iSeries <-> SecurePay server

- In cases where further transactions are required against this credit card (e.g. Pre-authorisations, Settlements, Refunds on a customer order), the iSeries will directly call a SecurePay API to process this transaction utilising previously stored token & reference data.

Web Payment Handler Prerequisites

General prerequisites

Following are guidelines on the requirements for the windows server and websphere for Web Payment Handler.

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 doesn’t have to be dedicated physical server.
4. Space on the server required by Web Payment Handler (WPH) will be less than 300MB.
5. Iptor need a temporary admin account to configure and install the web app.

6. Notepad or Wordpad required for editing configuration files.
7. SSL Certificate is required.
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 SecurePay Payment Gateway. You may need to check with the Payment Gateway support team for specific IP addresses.

Websphere

1. Open Liberty Profile can be used which is a ‘light’ version that can be downloaded and utilised for free.

Firewall

The Payment Handler Web server will need both inbound & outbound access to the SecurePay servers. It will need outbound access to the iSeries, this is typically done via ports specified in Java WPH config file *Server.XML* and SQL ports.

Install SSL / TLS

It is better to install SSL for the web application to ensure cyber data is safe and secure; consult your IT support staff.

iSeries

Firewall

As with the Payment Handler Web Server, the iSeries also needs both inbound & outbound access to the SecurePay servers. The iSeries must allow inbound access from the Payment Handler Web server. This is typically done via ports specified in Java WPH config file *Server.XML* and SQL ports.

Install SSL / TLS

As with the Payment Handler Web Server, the iSeries also needs security layer to communicate with the SecurePay servers.

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 SecurePay HPS page.

Setup & Configuration

Business rules for SecurePay Payment Gateway must be setup with support from Iptor consultants.

Note: The following configurational setups & business rules has to be setup for SecurePay Payment Gateway & Payment Handler. 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.

SecureFrame

Payment token replaces sensitive card information in SecureFrame. SecureFrame will create payment token on initial authorisation and sale. Once the payment token has been created any consequent transactions for the same user will use existing DirectPost in IP1 to post those transactions to SecurePay. The payment token will be stored in IP1 server.

To understand how SecureFrame interacts with third party landing page or web service, check:

https://www.securepay.com.au/wp-content/uploads/2017/06/SecureFrame_Integration_Guide.pdf

To understand how DirectPost works with IP1, check:

https://www.securepay.com.au/wp-content/uploads/2017/06/Direct_Post_Integration_Guide.pdf

Create test account

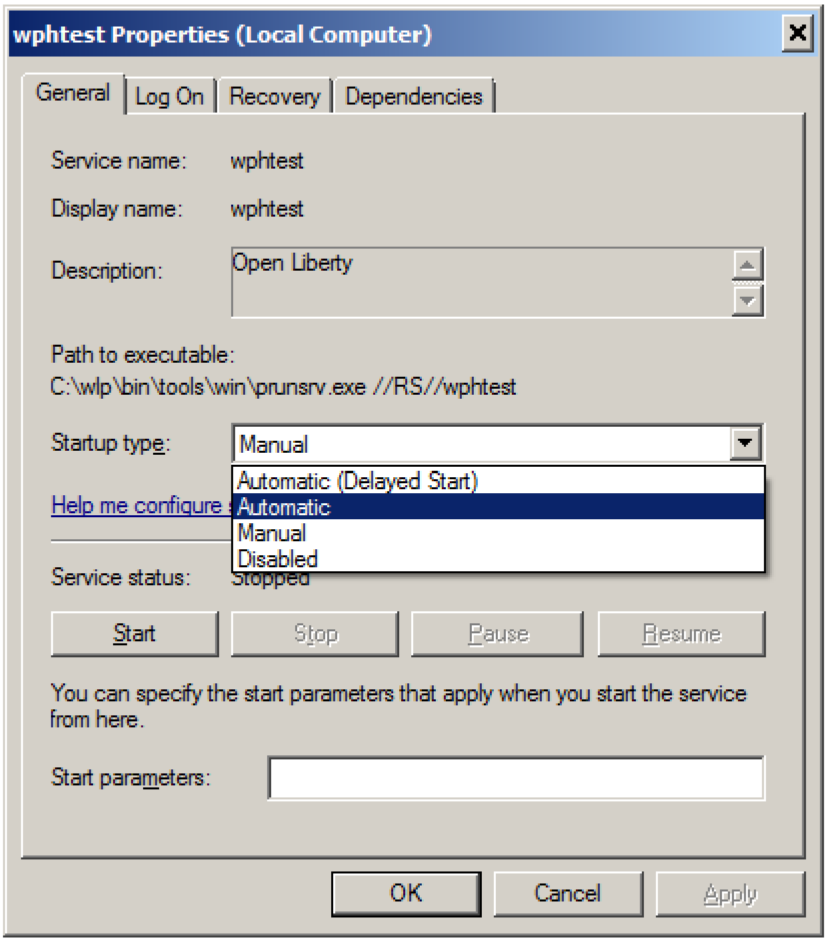
A merchant account must be created by SecurePay, via their sales team.

To create a merchant account go to https://www.securepay.com.au/about/contact-us/ Find the phone number under Sales Team, and give them a call.

Install the Payment Handler

Iptor will supply a deployment package, follow the instructions below to install the Payment handler web application.

1. Unzip wlp.zip file to the relevant drive resulting in folder like C:\wlp
2. Create test server and live server (*wphtest* and *wphlive* for example)
   - 2.1. Open command prompt as administrator and change directory to C:\wlp\bin
   - 2.2. Create test server - **server create server_name** (server_name is *wphtest* for example)
   - 2.3. Create Windows service - server registerWinService server_name
   - 2.4. Using Windows Services configuration tool set desired running mode:



Service name: wphtest

Display name: wphtest

Description: Open Liberty

Path to executable:
C:\wlp\bin\tools\win\prunsrv.exe //RS//wphtest

Startup type: Manual

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters:

2.5. Repeat previous steps for the live server (server_name is *wphlive* for example)

3. Install web application

3.1. Copy files *server.xml* and *cybs.wsdd* from WebApp.zip into newly created folders C:\wlp\usr\servers\wphtest and C:\wlp\usr\servers\wphlive. Override existing files if necessary.

3.2. In both servers edit file *server.xml* and update iSeries connection properties to the relevant ones, like:

```
<properties.db2.i.toolbox dateFormat="iso" naming="system" password="chairfan5"
serverName="IBSBKDEV" timeFormat="iso" user="IBSWDB79D"/>
```

3.3. In the test server file *server.xml* change ports to the values different from the live server (80 and 443), like:

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

3.4. Copy folder jph.war from WebApp.zip to C:\wlp\usr\servers\server_name\dropins

3.5. Review relevant payment system properties files in the folder C:\wlp\usr\servers\server_name\dropins\jph.war\WEB-INF\classes

3.6. If necessary, copy folder jph.war to the 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.

3.7. Start test server (or both servers) how described in section ‘Restart the Server’.

4. Configuration network connection
    4.1. Ports listed in files server.xml should be opened and https connectivity configured to the relevant payment system.
    4.2. SQL ports should be opened to iSeries (default ports are – 449,8470,8471,8475,8476).
    4.3. See section ‘iSeries configuration’ for the user profile and SQL stored procedure requirements.

Web application configuration

Server XML

Server.XML holds the information that allows communication between the Java WPH and IP1 environment(s). It replaces the environment.config file in .NET version.

Critical element to be configured is the <dataSource>. Check for the following:

- Points to the relevant iSeries
- User ID / password
- Libraries (this is optional, but if left out, then relies on the user id having JOBD with appropriate initial LIBL)

```
<server description="dev">
    <!-- Enable features -->
    <featureManager>
                                        <feature>servlet-4.0</feature>
                                        <feature>jsp-2.3</feature>
        <feature>jaxws-2.2</feature>
        <feature>jdbc-4.2</feature>
        <feature>jndi-1.0</feature>
<!--                                    <feature>mpHealth-1.0</feature> -->
<!--                                    <feature>localConnector-1.0</feature> -->
        <feature>localConnector-1.0</feature>
    </featureManager>
    <!-- For the keystore, default keys are generated and stored in a keystore. To provide the keystore
password, generate an
         encoded password using bin/securityUtility encode and add it below in the password attribute of
the keyStore element.
         Then uncomment the keyStore element. -->
    <keyStore id="defaultKeyStore" password="Liberty"/>

    <!--For a user registry configuration, configure your user registry. For example, configure a basic user
registry using the
        basicRegistry element. Specify your own user name below in the name attribute of the user
element.

For the password,
        generate an encoded password using bin/securityUtility encode and add it in the password attribute
of
the user element.
        Then uncomment the user element. -->
                     <!--<basicRegistry id="basic" realm="BasicRealm">
        <user name="bob" password="passw0rd" />
                     </basicRegistry>-->
                     <!-- <administrator-role>
                     <user>bob</user>
    </administrator-role> -->
    <!-- To access this server from a remote client add a host attribute to the following element, e.g. host="*" -->
    <httpEndpoint host="*" httpPort="80" httpsPort="443" id="defaultHttpEndpoint">
        <tcpOptions soReuseAddr="true"/>
    </httpEndpoint>
    <!-- Automatically expand WAR files and EAR files -->
    <applicationManager autoExpand="true" startTimeout="60s" stopTimeout="60s"/>
    <applicationMonitor updateTrigger="mbean"/>
    <jdbcDriver id="jtDrv" javax.sql.DataSource="com.ibm.as400.access.AS400JDBCDataSource" libraryRef="jt400">
    </jdbcDriver>
    <library id="jt400">
         <fileset dir="${shared.resource.dir}" includes="*.jar"/>
    </library>
    <!-- java:comp/DefaultDataSource -->
    <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="chairfan5" serverName="IBSBKDEV" timeFormat="iso" user="IBSWDB79D"/>
    </dataSource>
     <logging consoleLogLevel="INFO"/>
</server>
```

Secpay.properties

This holds the information that allows communication between Java WPH and SecurePay. It replaces the web.config file in .NET version.

```
# values - test or live (change to env=live if running live)
env=test
env.test.merchant.id=ABC0001
# allow to override default (jdbc/jph) for the given env.
env.test.jndi.79=jdbc/jph
# Secure Pay settings
env.test.SPpwd=abc123
env.test.SPurl=https://test.payment.securepay.com.au/secureframe/invoice
#default IP1 environment
env.test.SPenv.dft=79
env.test.SPpage.title=Iptor Payment

env.test.SPpage.header.image=https://iptor.com/wp-content/uploads/2016/08/Iptor-Full-Name-Logo-Colour-TM.png
env.test.SPpage.footer.image=https://www.securepay.com.au/wp-content/themes/securepay/images/logo.svg
env.test.SPconfirmation=yes
# values responsive, default and iframe?
env.test.SPtemplate=responsive
# bar | separate values from VISA, MASTERCARD, AMEX, DINERS, JCB, PAYPAL
# for no selection comment out or set to blank
env.test.SPcard.types=VISA|MASTERCARD|AMEX|DINERS|JCB|PAYPAL
env.test.SPzero.amt=true
env.live.merchant.id=Iptor
# allow to override default (jdbc/jph) for the given env.
env.live.jndi.79=jdbc/jph
# Secure Pay settings
env.live.SPpwd=abc123
env.live.SPurl=https://payment.securepay.com.au/secureframe/invoice
#default IP1 environment
env.live.SPenv.dft=79
env.live.SPpage.title=Iptor Payment
env.live.SPpage.header.image=https://iptor.com/wp-content/uploads/2016/08/Iptor-Full-Name-Logo-Colour-TM.png
env.live.SPpage.footer.image=https://www.securepay.com.au/wp-content/themes/securepay/images/logo.svg
env.live.SPconfirmation=yes
# values responsive, default and iframe?
env.live.SPtemplate=responsive
# bar | separate values from VISA, MASTERCARD, AMEX, DINERS, JCB, PAYPAL
# for no selection comment out or set to blank
env.live.SPcard.types=VISA|MASTERCARD|AMEX|DINERS|JCB|PAYPAL
env.live.SPzero.amt=true
```

After applying the changes, stop and restart the server. See next section.

Restart the server

Instructions to restart services:

- There two ways to do it as outlined below
- For either method, replace *server_name* as appropriate. i.e.
- *Server_name* = wphtest or wphlive

Method 1/

1. Stop relevant Windows service (wphtest or wphlive)
2. 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.
3. Unfortunately, Windows services option ‘restart’ can hang Java process quite badly, therefore it is necessary to do separate stop and start.

Method 2/

1. Open command prompt as Windows administrator.
2. CD c:\wlp\bin (or relevant folder where Open Liberty bin folder is).
3. Use the following command to stop the server:

   server stopWinService *server_name*
4. 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.
5. Use the following command to start the server:

   server startWinService *server_name*

iSeries configuration

User profile

User profile specified in the file server.xml should have JOBD with the following:

- TMSBASE/IBSBASE
- The base object library containing XAX000N e.g. TMSOBJZ79
- Another object library if XAS010 is elsewhere.
- The data library containing XACTD00P e.g. TMSDTA79P.

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!

```
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 TMSOBJZ79/XAX000N
```

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

Java Agent

1. Install Java Agent. For opening landing page URL, XAO636A uses Java Agent.

XAO636

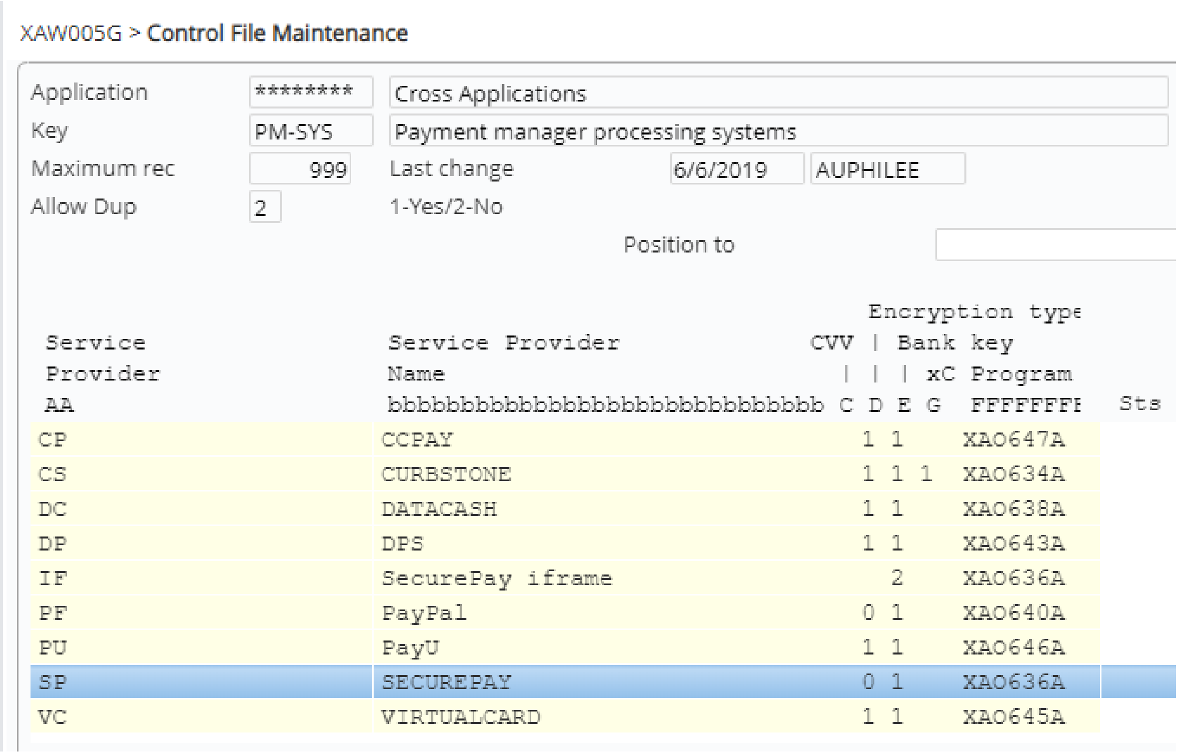
1. Deploy RPG program XAO636 if not already deployed.

IP1 Control files//Merchant details maintenance

1. Create/update the following control files entries.

   a. *********/PM-SYS** Payment manager processing system

      – include entry for SecurePay as below.



XAW005G > **Control File Maintenance**

Application ******** Cross Applications

Key PM-SYS Payment manager processing systems

Maximum rec 999 Last change 6/6/2019 AUPHILEE

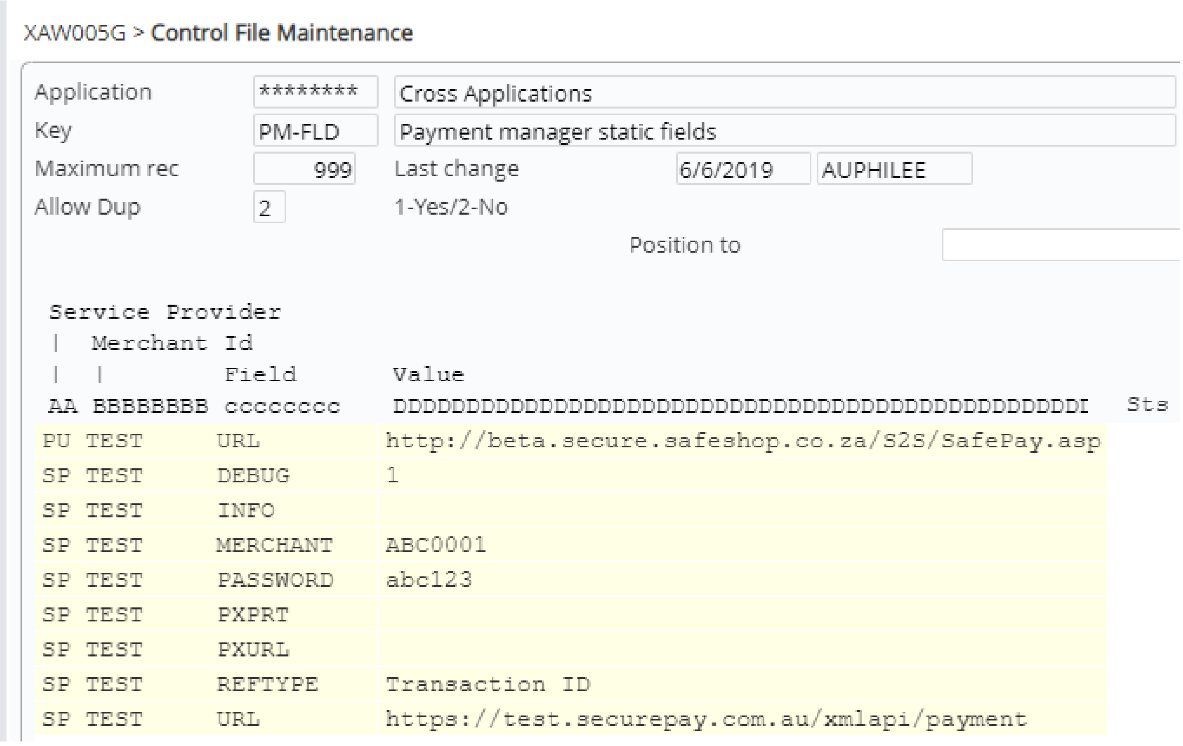
Allow Dup 2 1-Yes/2-No

Position to

```
                                                         Encryption type
Service                 Service Provider             CVV | Bank key
Provider                Name                           | | | xC Program
AA                      bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb C D E G  FFFFFFFF  Sts
CP                      CCPAY                            1 1    XA0647A
CS                      CURBSTONE                        1 1 1  XA0634A
DC                      DATACASH                         1 1    XA0638A
DP                      DPS                              1 1    XA0643A
IF                      SecurePay iframe                   2    XA0636A
PF                      PayPal                           0 1    XA0640A
PU                      PayU                             1 1    XA0646A
SP                      SECUREPAY                        0 1    XA0636A
VC                      VIRTUALCARD                      1 1    XA0645A
```

b. **********/PM-FLD** Payment manager static fields

- setup the various required fields for SecurePay



XAW005G > **Control File Maintenance**

Application ******** Cross Applications

Key PM-FLD Payment manager static fields

Maximum rec 999 Last change 6/6/2019 AUPHILEE

Allow Dup 2 1-Yes/2-No

Position to

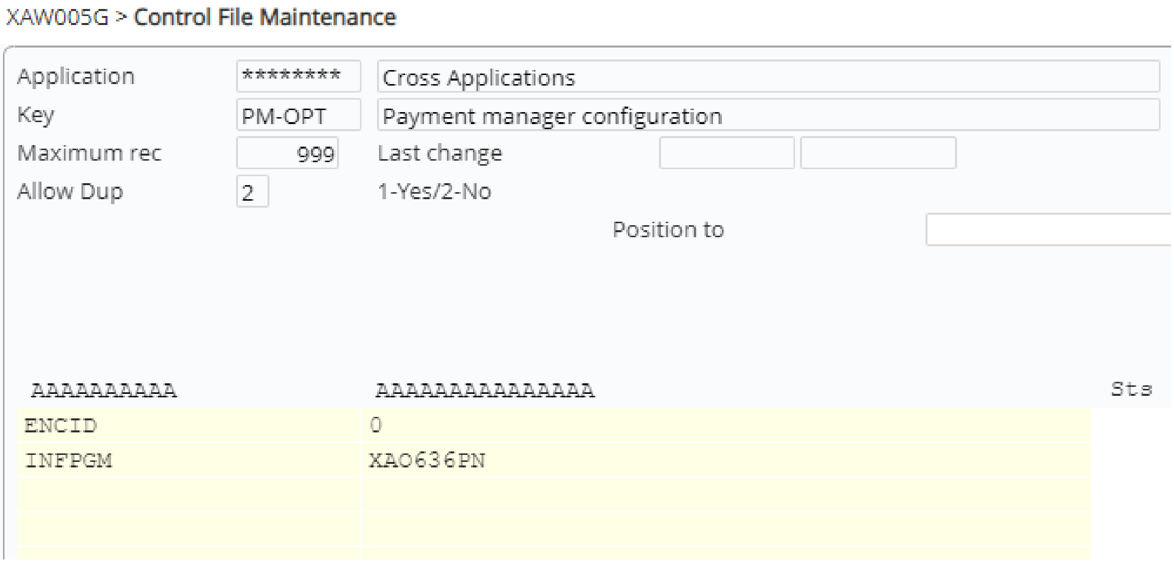
```
Service Provider
|  Merchant Id
|  |        Field       Value
AA BBBBBBBB cccccccc    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD  Sts
PU TEST     URL         http://beta.secure.safeshop.co.za/S2S/SafePay.asp
SP TEST     DEBUG       1
SP TEST     INFO
SP TEST     MERCHANT    ABC0001
SP TEST     PASSWORD    abc123
SP TEST     PXPRT
SP TEST     PXURL
SP TEST     REFTYPE     Transaction ID
SP TEST     URL         https://test.securepay.com.au/xmlapi/payment
```

Field settings required for SecurePay:

Field	Description
DEBUG	Universal; can have the value of 1 or 2 only, even though there is no validation on it. The transaction will process for both values but option 1 gives additional technical data on the transaction being processed, for investigational purposes. Logs comments in IFS temporary folder.
LIVE	Not used any more
MERCHANT	Merchant ID for SecurePay gateway
PASSWORD	Password to access SecurePay gateway.
PXPRT	Proxy port - provided by the customer.
PXURL	URL for the Proxy user – provided by the customer.
REFTYPE	Defines the label on the hosted pages for the Primary Reference field
URL	SecurePay URL
URLL	URL for the Payment Handler landing page.
URLT	SecurePay URL for triggered payment transactions
VLDAMT	universal; Pre-authorised amount for c/c and expiry date validation via a dummy transaction. Can be overwritten. For US its 0.
VLDAUTH	the number of days/hours to hold the pre-authorisation. Format is 1DDHH, '1'-Yes 'DDHH'-days/hrs from auth date.
WAIT	universal; time in seconds for iSeries to wait for landing page.

c. **********/PM-OPT** Payment manager configuration

- Set INFPGM to XAO636PN
- ENCID must always be set to '0'



XAW005G > Control File Maintenance

Application	********	Cross Applications
Key	PM-OPT	Payment manager configuration
Maximum rec	999	Last change
Allow Dup	2	1-Yes/2-No

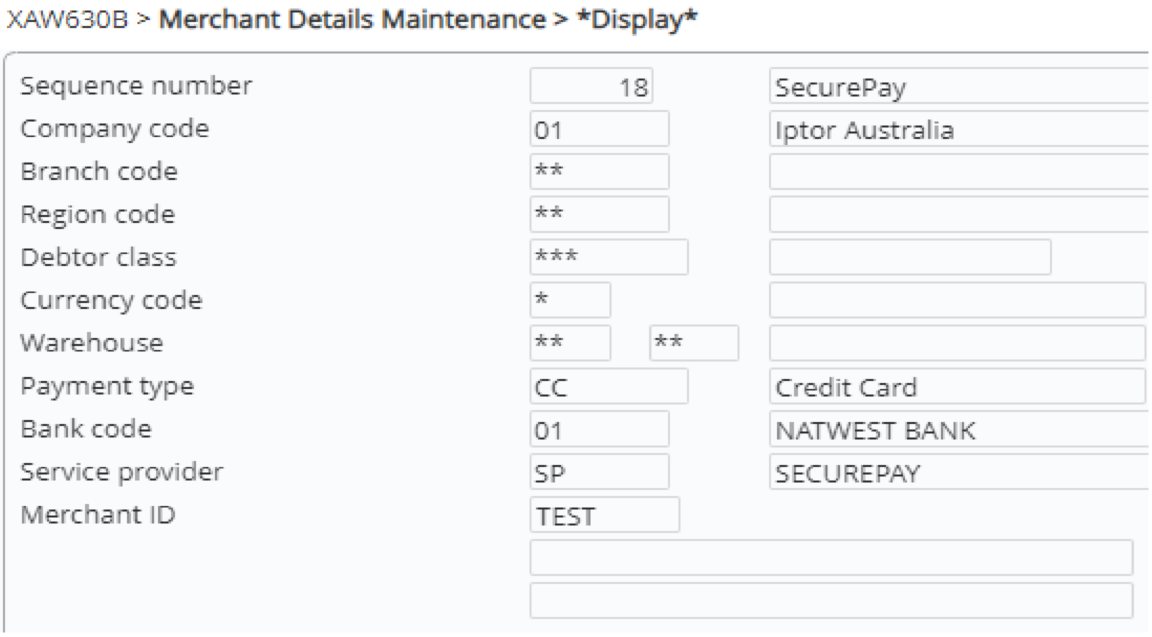
Position to

AAAAAAAAAA	AAAAAAAAAAAAAA	Sts
ENCID	0	
INFPGM	XAO636PN	

2. Configure Merchant detail maintenance (XAW630); use relevant SecurePay account entries as per ********/PM-FLD.

IMPORTANT NOTE:

- Make sure you include a ‘catch all’ rule that links with a valid SecurePay provider regardless of company, currency, payment type, etc.
- This is required to ensure F10=Confirm using an existing card that has already been linked to an order will work properly during order maintenance or Work with Failed Credit Card payment entry.



XAW630B > **Merchant Details Maintenance > *Display***

Sequence number	18		SecurePay
Company code	01		Iptor Australia
Branch code	**		
Region code	**		
Debtor class	***		
Currency code	*		
Warehouse	**	**	
Payment type	CC		Credit Card
Bank code	01		NATWEST BANK
Service provider	SP		SECUREPAY
Merchant ID	TEST		

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

Iptor IP1 Web Services

Customers using Iptor IP1 Web Services can change web site to interact with SecurePay landing page and pass appropriate data back to IP1 through web services (**Customer/Web Developer**).

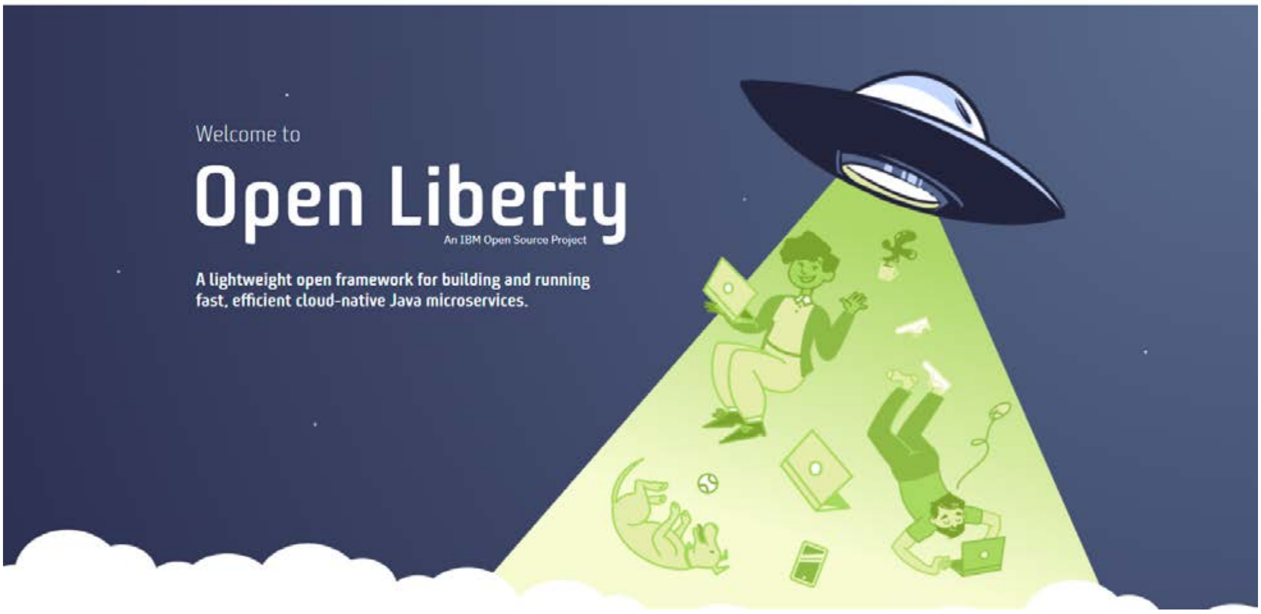
1. Refer to "SecureFrame Integration Guide" on SecurePay web site for list of available return fields (section 4.3)
2. Note that Iptor does NOT support Fraud Guard, or Surcharge facilities outlined in the SecureFrame guide.

For further configuration notes and field mapping for Iptor IP1 Web Services refer to SecurePay (SecureFrame) Token Field Mapping v1.3.xlsx supplied by Iptor Australia.

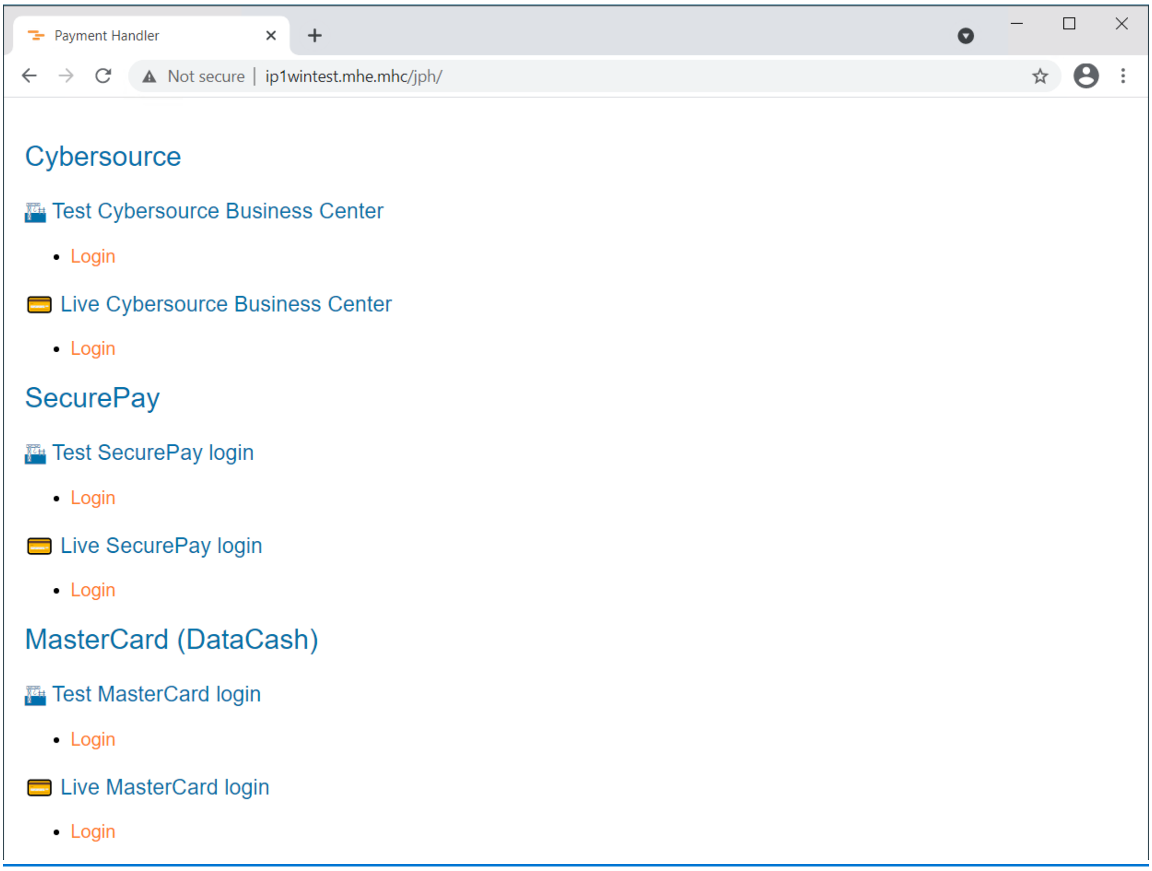
Testing

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:


Welcome to
Open Liberty
An IBM Open Source Project
A lightweight open framework for building and running fast, efficient cloud-native Java microservices.

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
   b. 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.



Cybersource

Test Cybersource Business Center

- Login

Live Cybersource Business Center

- Login

SecurePay

Test SecurePay login

- Login

Live SecurePay login

- Login

MasterCard (DataCash)

Test MasterCard login

- Login

Live MasterCard login

- Login

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 secpay.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 4111111111111111
            2. This test verifies:

a. All being well, should bring up the SecurePay 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 & SecurePay, 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 SecurePay 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.

Troubleshooting

1. 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
2. If URL brings up **error 404 – File or directory not found** error on browser.
    a. Check all installation steps have been completed on both windows and iSeries.
    b. Check that the relevant windows service is running.
    c. You may need to include specific port# in the ********/PM URL* field settings, rather than relying on default.
    d. 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
        ```

    e. 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.
3. 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.

4. 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.

5. 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 log file - XAPCD00P.

Appendix

FAQ – Web Payment Handler (WPH) prerequisites

a) Do we need separate installation & configuration for each Merchant Id?
Currently Iptor has only one deployment file which can be used for all supported Payment Gateway merchants. However, each merchant will require its own customised configuration file specifying client preferences and merchant id 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 suggests 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.

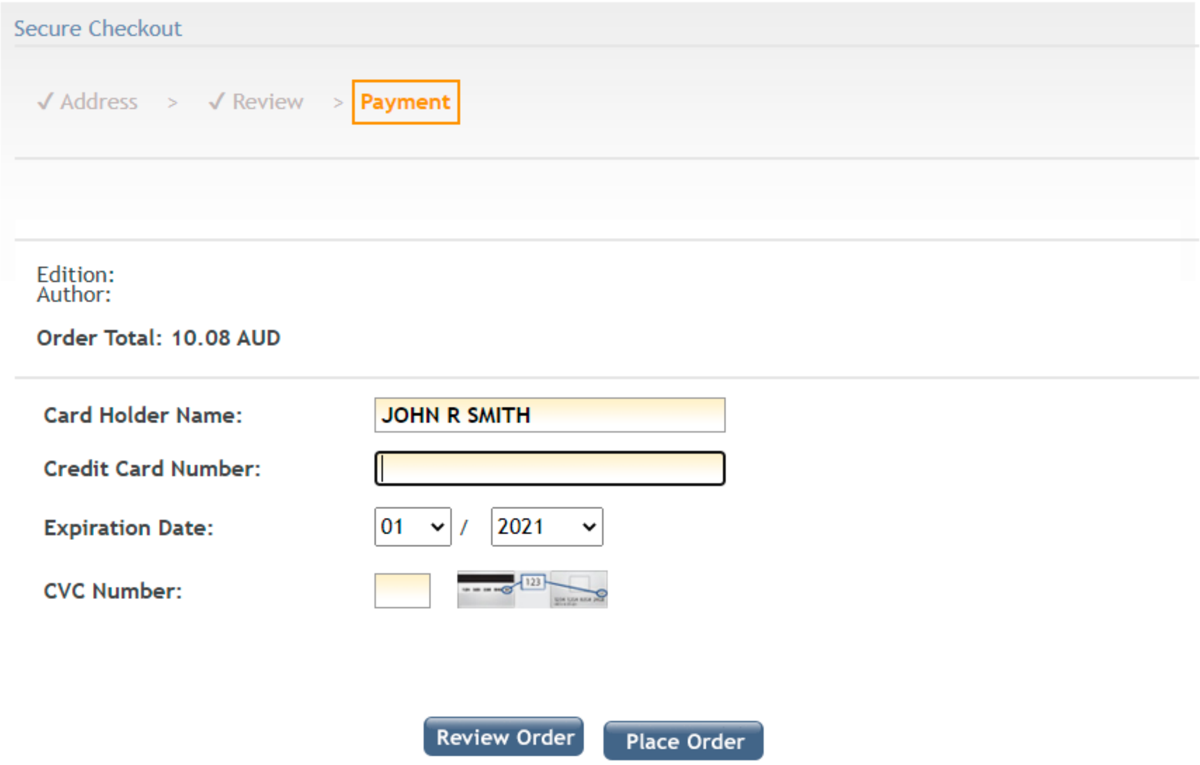
However, clients may prefer to install multiple copies of the application onto one server, or onto multiple servers.

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.

Sample Data Entry Web Page

Sample Data Entry web page used for Credit Card entering.



Secure Checkout

✓ Address > ✓ Review > Payment

Edition:
Author:

Order Total: 10.08 AUD

Card Holder Name:	JOHN R SMITH
Credit Card Number:	
Expiration Date:	01 / 2021
CVC Number:	

Other Tips

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

1. Wintel setup
   a. As per Install the Payment Handler section in this document.
2. 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
3. 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 SecurePay 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.
         From
         http://dnsservername /ph/SPGateway
         to
         http://dnsservername /jph/SPGateway

UAT test plan suggestions

1. There are NO changes in secondary transaction processing with the migration from .Net to Java WPH. This is all handled directly between IP1 and SecurePay servers, and there are no program changes on IP1 side at all.
2. Therefore, key requirement is simply to test the initial capture of card details via the SecurePay Landing Page.
3. Only reason to check secondary transaction would be to ensure Java version has captured back correct token data into IP1 for secondary processing.

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 SecurePay provider and swap the merchant id from LIVE to TEST.
   c. If you have multiple merchants, you may have something like xxTEST or xxLIVE merchant's 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 SecurePay card payments, then you will need to:
2. Wintel server
    a. Add new IP1 environment entries into the secpay.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 MAP.

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 HPS landing page with merchant (pageset)

1. Iptor can assist with providing sample html pages.
2. Send relevant html to Datacash support, asking them to link that with the particular merchant id (vtID)
3. They will provide a pageset number that they have allocated this html layout against for that merchant id (vtID)
4. Edit Dc.properties file(s) in test and/or live wintel servers to set env.live.DCpgs.setting. eg. Following sets the page set to 3933 for test merchant id 99007724
    a. env.test.DCpgs.99007724=3933
5. Note:
    a. Be careful if modifying the html layout that you don't change the field number settings, as these are important to how our WPH s/w passes appropriate information to Datacash. Our default html page uses field number settings:

DCCapfNam	=1
DCCapfCns	=7
DCCapfEnv	=8
DCCapfMch	=9

New Merchant ID

1. Ensure appropriate HPS landing page has been loaded with Datacash for the merchant id (vtID) and pageset number allocated
2. You will also need to know the password from Datacash for this vtID

3. Wintel servers
    a. Dc.properties
        i. If merchant id is not already in the file, then copy and modify from one of the existing merchants
        ii. Change the .nnnnnnnn portion to new merchant id
        iii. Set DCpwd to the new password for this vtID
        iv. Set DCpgs to the page set number that has been allocated to this vtID
        v. Double check each of the DCCap* settings (see notes in prev section about field numbers)