cardconnect®

SnapPay Installation Instructions

XXSNP Application Components

Author:	Jeffrey Swanson
Creation Date:	1 January 2019
Last Updated:	17 December 2019
Version:	1.0

Contents

Installation Instructions for XXSNP Application

Environment Name:	Machine:

Client and Environment Requirements

NOTE: ***The below instructions and information assumes that the client has taken all the necessary steps and actions to configure their EBS System to accept credit card transactions. This includes and is not limited to the following:***

- ***Credit Card specific receipt and payment methods as well as appropriately setting the OM: Credit Card Privileges profile option. Please see your AR, Order Management, and Oracle Payments user and implementation guides for further information on how to properly configure your EBS system to accept credit cards.***

Pre-Installation Steps

Pre-Installation Checklist

- ❑ Verify $APPL_TOP exists
- ❑ Verify that AdSplice is installed and working properly

Single Tier XXSNP Installation

Deploy the SnapPay Installation shell script (UNIX as applmgr)

A shell script will (1) create the $XXSNP_TOP path, (2) XXSNP database user, (3) XXSNP and APPS database objects, (4) deploy the media/JSP files for the web tokenizer, (5) upload ldt files that will create concurrent programs, associated value sets and user profile options, and (6) create the payment system and payee for gateway model processing. To execute the shell script successfully, you will need the following information:

Information Needed	Reason
APPS username/password	Query the database and compile database objects via SQLPlus
SYSTEM password	Used during AdSplice and AutoConfig
Oracle Database SID	Query the database and compile database objects via SQLPlus
SnapPay API URL	Used to setup the SnapPay payment system (i.e. https://SnapPayGlobal.com/REST/api/)
SnapPay web tokenizer URL	Used to setup the SnapPay web tokenizer (i.e. https://SnapPayGlobal.com/interop/interoprequest?reqno=)

SnapPay Account ID	Will be attached to the SnapPay payment system and defaulted on the SnapPay payee
SnapPay HMAC Key	All API calls sent out to SnapPay require HMAC authentication
Enable SnapPay forms personalizations	The forms personalizations necessary to utilize the SnapPay web tokenizer will be deployed. Based on the business, you may automatically have this form personalization enabled or disabled.

- ❑ Stop your environment so that only one result is returned when the command ps -ef | grep -i fnd is ran. This includes running adstpall.sh and shutting down concurrent managers.
- ❑ Create custom application directory structure under $APPL_TOP
    1. ftp the xxsnp.zip file to /tmp/
    2. Navigate to $APPL_TOP
    3. Create XXSNP_TOP application directory structure by unzipping supplied xxsnp.zip under $APPL_TOP:

       unzip /tmp/xxsnp.zip
- ❑ Execute the XXSNP Installation shell script with the below command.
    - $APPL_TOP/xxsnp/12.0.0/install/shellScript/xxsnp_main.sh
- ❑ Regenerate your environment
    - cd $APPL_TOP
    - . <SID>_<hostname>.env RUN
        - Replacing <SID> and <hostname> with the environment's respective values
- ❑ Bring up your EBS application

Multiple Node XXSNP Installation

Prepare the environment for deployment (UNIX as applmgr)

- ❑ Create custom application directory structure under $APPL_TOP
    1. ftp the xxsnp.zip file to /tmp/
    2. Navigate to $APPL_TOP
    3. Create XXSNP_TOP application directory structure by unzipping supplied xxsnp.zip under $APPL_TOP:

       unzip /tmp/xxsnp.zip
- ❑ Shut down all application services (i.e. WebLogic, each oacore node)

Add the SnapPay payment system servlet to the oacore template (UNIX as applmgr)

Before installing the SnapPay application, the payment system servlet will be added to the template so it can be cascaded during autoconfig (ran during the next step of this installation). When prompted by the script you will need the following value:

Information Needed	Reason
oacore_web_xml_FMW.tmp	The script requires the specific file name that will be edited

- ❑ $APPL_TOP/xxsnp/12.0.0/install/shellScript/xxsnp_add_servlet_config.sh

At the conclusion of this script, you should see the following entires in the $FND_TOP/admin/template/oacore_web_xml_FMW.tmp file:

```
<!-- SnapPay Payment System Servlet -->

        <servlet>
                <servlet-name>SnapPayServlet</servlet-name>
                <servlet-class>com.snappay.ebs.ipayment.SnapPayServlet</servlet-class>
                <init-param>
                        <param-name>debug</param-name>
                        <param-value>true</param-value>
                </init-param>
        </servlet>

<!-- IBY Servlet -->

     <servlet>
     <servlet-name>ibyecapp</servlet-name>
     <servlet-class>oracle.apps.iby.ecservlet.ECServlet</servlet-class>
     </servlet>

     <servlet>
     <servlet-name>LoopBackServlet</servlet-name>
     <servlet-class>oracle.apps.iby.bep.loop.LoopBackServlet</servlet-class>
       <init-param>
         <param-name>debug</param-name>
         <param-value>true</param-value>
       </init-param>
    </servlet>

   <servlet>
   <servlet-name>LoopProcServlet</servlet-name>
   <servlet-class>oracle.apps.iby.bep.proc.loopproc.LoopProcServlet</servlet-class>
    <init-param>
      <param-name>MAX_ARCHIVE_AGE</param-name>
      <param-value>1</param-value>
    </init-param>
 </servlet>

  <servlet>
  <servlet-name>TransmitServlet</servlet-name>
  <servlet-class>oracle.apps.iby.bep.TransmitServlet</servlet-class>
   <init-param>
     <param-name>MAX_ARCHIVE_AGE</param-name>
     <param-value>7</param-value>
   </init-param>
 </servlet>

<!-- SnapPay Payment System Servlet Mapping -->

        <servlet-mapping>
                <servlet-name>SnapPayServlet</servlet-name>
                <url-pattern>/oramipp_snp/*</url-pattern>
        </servlet-mapping>

<!-- IBY Servlet Mapping -->
```

Note: The script will create a backup template in the form of oacore_web_xml_FMW.tmp.bak_YYYYMMDDHHMISS

Deploy the SnapPay installation via AdSplice (UNIX as applmgr)

A series of shell scripts will be needed to successfully install the XXSNP application across multiple application tiers.

Information Needed	Reason
APPS username/password	Query the database and compile database objects via SQLPlus
SYSTEM password	Used during AdSplice

- ❑ Execute the following shell script for AdSplice to run on the first node
    - o $APPL_TOP/xxsnp/12.0.0/install/shellScript/xxsnp_adsplice.sh
- ❑ Per Metalink doc 1577707.1: *"For shared APPL_TOP installations, run AutoConfig for the remaining nodes (there is no need to rerun AD Splice). Or, if the APPL_TOP is not shared, repeat Steps 8 to 12 in Section 1 [of 1577707.1] on all the other nodes."*

Regenerate your environment file

For future scripts in this session to recognize the newly created application and environment variable (XXSNP_TOP), the environment file will need to be regenerated.

- ❑ cd $APPL_TOP
- ❑ . <SID>_<hostname>.env RUN
    - o Replacing <SID> and <hostname> with the environment's respective values

Checkpoint

The following command should be used to check if there is one result which points to the XXSNP directory

env | grep XXSNP

Deploy the SnapPay Java code (UNIX as applmgr)

Execute the following shell script on each node for the SnapPay JAR and JSPs to visible by each

- ❑ $APPL_TOP/xxsnp/12.0.0/install/shellScript/xxsnp_java_deploy.sh

Deploy the SnapPay database objects (UNIX as applmgr)

Execute the following shell script on one node for all the AOL and database objects to be complied:

Information Needed	Reason
APPS username/password	Query the database

SYSTEM password	Used during adadmin
Oracle Database SID	Query the database
SnapPay API URL	Used to setup the SnapPay payment system (i.e. https://SnapPayGlobal.com/REST/api/)
SnapPay web tokenizer URL	Used to setup the SnapPay web tokenizer (i.e. https://SnapPayGlobal.com/interop/interoprequest?reqno=)
SnapPay HMAC Key	All API calls sent out to SnapPay require HMAC authentication
SnapPay Account ID	Will be attached to the SnapPay payment system and defaulted on the SnapPay payee

- ❑ $APPL_TOP/xxsnp/12.0.0/install/shellScript/xxsnp_aol_and_db.sh

Start application services

Bring up the application services if they have not been restarted since being taken down for the execution of AdSplice.

CardConnect iFrame Installation

Execute the CardConnect iFrame shell script (UNIX as applmgr)

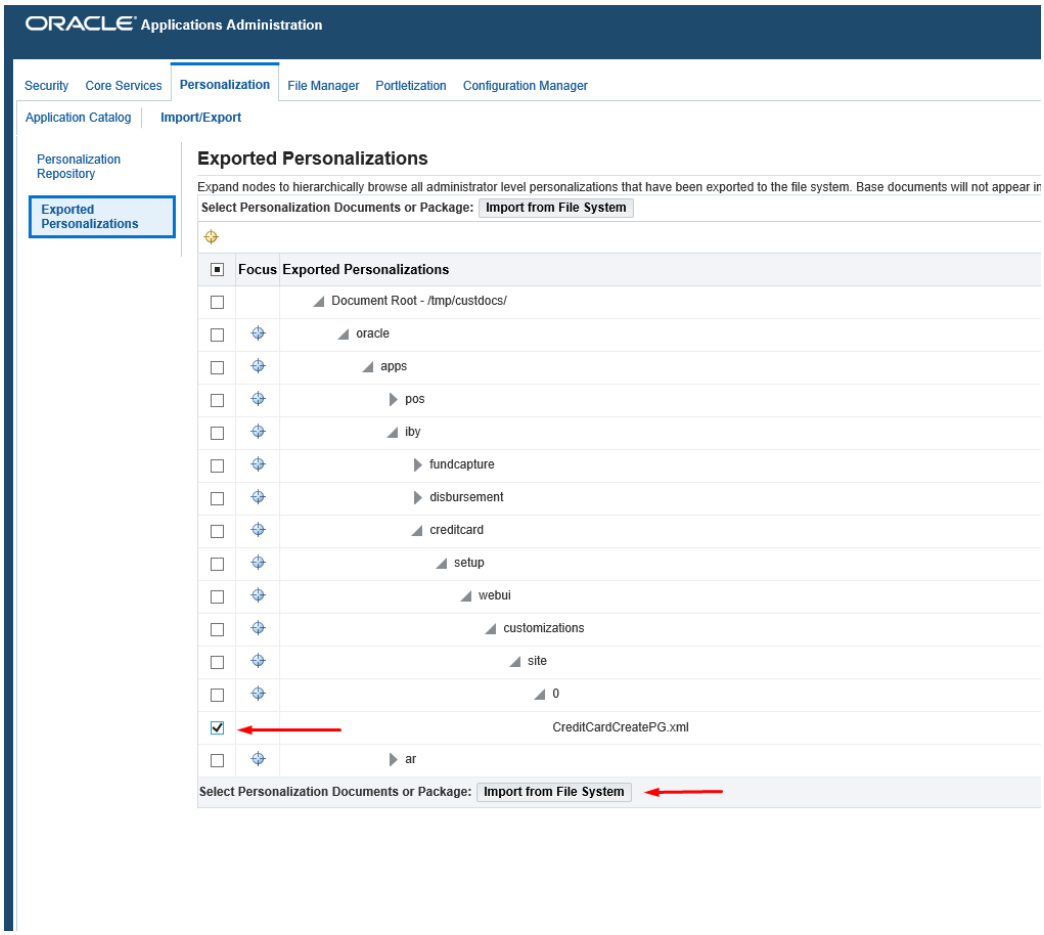
A shell script will be used to extract the tar file of XML files used to upload the personalization to the credit card web page on the customer standard. To execute the shell script successfully, you will need the following information:

Information Needed	Reason
APPS username/password	Query the database
Oracle Database SID	Query the database
CardConnect site	To build out the iFrame URL. This value will be provided by your SnapPay implementation team
Method of entry of credit card information	The iFrame will need to be configured if SRED devices or keyboards will be used to enter the PAN information

- ❑ $APPL_TOP/xxsnp/12.0.0/install/shellScript/xxsnp_iFrame.sh

Upload the OAF personalization

Under the Functional Administrator responsibility, navigate to “Personalization” > “Import/Export” > “Exported Personalizations” and select the file under the directory of “oracle/apps/iby/creditcard/setup/webui/customizations/site/0/CreditCardCreatePG.xml”. Click on “Import from File System” and the OAF personalization should take care of the iFrame setup document under the customer standard page. If other pages are used, please refer to the “IFRAME Personalizations (Generic) for iReceivables Customer Standard and Service.doc” document.



Sysadmin Installation

Set System Profile Options (System Administrator -> Profile-> System)

Note: The following profile options encompass all SnapPay integrations. Depending on how the xxsnp_aol_and_db.sh script is executed, you may not have all of these profile options in your environment.

Shared Options

Profile option	Recommended value	Description
XXSNP: Communication Method	HTTP	Non-editable profile option
XXSNP: Customer AccountID		SnapPay account number. This value will be provided by your project team
XXSNP: Enable CLOB Audit Tables	Yes	Responses from the SnapPay APIs can be stored in a custom table
XXSNP: Enable Verbose Log Messages	Yes	Concurrent programs provided by SnapPay can add extra messages to the log file
XXSNP: Force TLS Version	TLSv1.2	Forces the TLS version to avoid having to make JVM start-up arguments. If the Java version is up to date, it is still recommended to be set at the highest level to ensure no issues with WebLogic
XXSNP: Send Customer Info With Token	Yes	Will pull the customer information to send with the tokenization request. This will eliminate the need to populate some of the required tokenization fields regarding billing information to save the token

XXSNP: SnapPay Authentication Type	BOTH	API authentication can be set to either HMAC or BOTH (BASIC is only for backward compatibilty issues and should not be used only if strictly instructed by the implementation team)
XXSNP: SnapPay Web Tokenizer UserID		SnapPay can be configured to either use a P2PE device or keyboard entry for PAN tokenization. This value will be provided by your implementation team
XXSNP: Status email		Profile option used in SnapPay concurrent programs

PaymentSystem Options

Profile option	Recommended value	Description
XXSNP: Convert Batch Size	1000	If clear PAN exists from previous payment system provider, our concurrent program can tokenize the credit card numbers so Oracle does not house this information
XXSNP: Payment System URL		Used by the CardConvert concurrent program. This value will be provided by the implementation team
XXSNP: SnapPay User ID		This value will be provided by your implementation team
XXSNP: Tokenizer Registration Level		Dictates on where the credit card is assigned at the time of tokenization

EIPP Options

Profile option	Recommended value	Description
XXSNP: Batch Source For Fees		If fees are in use, this value will hold the batch source the fees are created with
XXSNP: Customer Upload Change Log Days to Keep	90	Based on this value, the customer upload program log table will hold X number of days of uploaded data
XXSNP: Fee GL Account		GL account associated with the fee
XXSNP: Invoice Upload Change Log Days to Keep	90	Based on this value, the invoice upload program log table will hold X number of days of uploaded data
XXSNP: Receipt Method		Receipt method that will be created when pulling the payment information from the previously uploaded invoices

AP Options

Profile option	Recommended value	Description
XXSNP: AP Payment Document		Specify the AP payment document associated with invoices which will be used to pay with SnapPay
XXSNP: AP Payment Profile		Specify the AP payment profile associated with invoices which will be used to pay with SnapPay

Prerequisite Patch Levels

To make the API calls successfully, all data should be transmitted via the TLS 1.2 communication protocol. Please review the following Oracle documents based on your environment to make sure the necessary patches and configurations are enabled:

Enabling TLS in Oracle E-Business Suite Release 12.2 (Doc ID 1367293.1)

Open and Closed Issues for this Deliverable

Open Issues

ID	Issue	Resolution	Responsibility	Target Date	Impact Date

Closed Issues

ID	Issue	Resolution	Responsibility	Target Date	Impact Date