DPS POS Integration Certification Request and Test Scripts

dps | paymentexpress

1 DOCUMENT HISTORY

Version	Author	Date
3.0.0	David Merry	01/2012
3.0.1	Grant Shannon	01/2012
3.0.2	David Merry	01/2012
3.0.3	James Rees	06/2013

2 POS DETAILS

2.1 VENDOR DETAILS

POS Vendor:

Technical Contact Name:

Phone Number:

Email:

Job Title:

Trading Name:

Other Contact Name:

Phone Number:

Email:

Job Title:

2.2 POS INFORMATION

2.2.1 POS Name and Version

POS Name:

Version Number:

2.2.2 Integration Method

Please state which integration the POS uses (e.g. ActiveX, XML interface):

Please give any details of complications in the integration (e.g. ActiveX wrappers):

Does the POS use DPS' dialogue boxes to display transaction messages?

2.2.3 Printing

Does the POS print the EFTPOS receipts, or does PX EFTPOS print EFTPOS receipts?

If the POS prints the EFTPOS receipts, please explain why:

What printers and printing systems does the POS support?

2.2.4 POS Environment

Will the POS be in an attended, semi-attended, or unintended environment?

If the POS will not be in an attended environment, please give brief details of any correspondence with DPS about this:

2.2.5 Feature support

☐ Refund	☐ Shift Totals	☐ Cash Out	☐ Tipping
☐ Hospitality	☐ Fuel Transactions	☐ Adv. Pur. Data	☐ Cheque
☐ Flybuys	☐ Read Card Method	☐ Multi merchant	☐ Gift / Loyalty Card
☐ Cheque	☐ Self Service Kiosk	☐ Manual PAN Entry	☐ Other

3 DETAILS OF TESTING

3.1 TESTING SCOPE

These tests are designed to provide some confidence that the POS will not present significant risks to the financial integrity or the usability of the Payment Express EFTPOS solution. The tests described in here should be considered *minimal,* and it is recommended that POS vendors undertake more thorough testing of their POS systems.

3.1.1 Out of scope

The following are among the significant areas not covered by these tests:

1. Long term integrity of stored financial records in the POS
2. Security of sensitive data stored by the POS
3. Any features of the POS outside of the EFTPOS integration
4. Elegance of the EFTPOS integration
5. Fuel transactions (to be discussed with DPS)

3.2 TESTING TIMELINES

DPS can provide timelines for testing the integration of a POS but it cannot provide timelines for a POS becoming certified. To ensure timely completion of the certification, it is recommended that POS developers work through this test script themselves before sending the POS to DPS.

If you believe that DPS has agreed to any timelines for the certification testing, please give details below:

3.3 TEST ENVIRONMENT SETUP

It is the POS vendor's responsibility to provide DPS' certification team with a testing environment in a timely manner. Our preference is for a preconfigured virtual machine to be provided to us. However if the setup of the POS is relatively easy, the POS vendor may provide installation files and documentation. If installation is quite involved, the POS vendor should set up a test environment for DPS which they may do by providing a physical machine or by sending a technician to DPS to undertake the installation.

Please note that we can only accept virtual machines that use Microsoft HyperV(.vhd) format.

Please give details of how the POS developer has assisted DPS with setup of the test environment. Make sure you have provided any login details. If DPS is to install please attach a separate document detailing the installation process:

4 REQUIREMENTS OF INTEGRATION

The testing process is intended to provide some confidence that the following requirements are met. If a test reveals that one of these requirements is not met, then, even if the POS meets all the expected outcomes for that test, it might fail the certification. If a customer using the POS finds that it does not meet one of these requirements, then the POS vendor should work with DPS to make sure that this is corrected.

Requirement	Description	When required
1	Messages provided by the POS give accurate information about transactions	Always
2	POS can recover the status of an incomplete transaction after a crash	Always
3	POS uses best practice for implementing the interface with the EFTPOS software	Always
4	The POS sends transactions that match what the user appears to have requested at the POS	Always
5	The POS prints a correct EFTPOS receipt for every transaction and inhibits transactions when the printer is offline.	POS handling printing
6	The POS provides whatever DPS sends it as merchant dialogues	POS providing dialogues
7	The POS provides all essential features	POS deployed in such a way that DPS' client cannot be used. The client must not be used to handle any transactions.

5 TEST CARDS

Test Card Name	PAN	Accounts	PIN
Visa Credit	4999 9999 9999 9109	Credit	1234
Visa Credit and Debit	4999 9999 9999 9108	Cheque and Credit	1234
MasterCard Credit	5999 9999 9999 9108	Credit	No
Debit Card	9999 9999 9999 9108	Savings	1234

6 TEST EXECUTION

- All test cases in section 7.1 should be executed for all POS.
- Section 7.2 only applies to POS that have been integrated via the XML interface.
- Section 7.3 test cases should only be executed for POS that have implemented custom dialogs.
- Section 7.4 test cases should only be executed for POS that control printers.
- The remaining test cases should only be executed if the POS has integrated the appropriate features. Otherwise they **are not applicable** to the certification.

7 SUMMARY OF TEST CASES

7.1 TEST CASES THAT APPLY TO ALL POS

7.1.1 Purchase transactions processed correctly

7.1.2 Non-Financial Transactions

7.1.3 Exception handling

7.2 TEST CASES THAT APPLY TO ALL POS USING XML INTEGRATION

7.3 TEST CASES THAT APPLY TO ALL POS PROVIDING CUSTOM DIALOGUES

7.4 TEST CASES THAT APPLY TO POS CONTROLLING PRINTERS

7.4.1 Test cases that apply to all POS controlling printers

7.4.2 Test cases that apply to all POS controlling printers PX EFTPOS cannot connect to

7.5 TIPPING TRANSACTIONS

7.6 HOSPITALITY TRANSACTIONS

7.7 CHEQUE TRANSACTIONS

7.8 FLYBUYS TRANSACTIONS

7.9 EPAY TRANSACTIONS

7.10 TEST CASES FOR COMMON FEATURES

7.10.1 Financial Transactions

7.10.2 Non-Financial Transactions

8 TEST CASES

8.1 CASES THAT APPLY TO ALL POS

8.1.1 Generic Cases

Case 1	**Purchase transaction approved with signature**
Rationale:	To check that a POS can process a transaction and correctly recognise the result.
Test steps:	1. Run a purchase transaction with test card Visa Credit 2. Do **not** enter a PIN when prompted. Just press enter. 3. Choose "yes" when prompted to accept signature
Expected results:	**For all POS:** **1.** The transaction runs successfully from end to end without any errors **2.** The POS recognises the transaction as approved **3.** There are no un-necessary commands such as 'DoReadCard' at the beginning of the transaction **4.** A receipt is printed for the customer to sign when the 'accept with signature yes/no' prompt appears, and before a choice has been made **5.** A second receipt is printed for the customer to take away after a choice has been made
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 2	**Purchase transaction approved with PIN**
Rationale:	To check that a POS can process a transaction and correctly recognise the result. A PIN transaction has a different transaction flow and response code from a signature transaction.
Test steps:	1. Run a transaction with test card Visa Credit 2. When prompted to enter a PIN, enter 1234 and press enter.
Expected results:	**1.** The transaction runs successfully from end to end without any errors **2.** The POS recognises the transaction as approved **3.** An EFTPOS receipt **must** be printed when the 'transaction approved' dialogue appears. Presenting an option for the customer to decline the receipt is acceptable.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 3	**Purchase transaction declined with signature**
Rationale:	To check that the POS correctly differentiates between approved and declined transactions.
Test steps:	1. Run a transaction with test card Visa Credit 2. At the account select stage, choose the credit account 3. When asked to enter a PIN, don't, but press enter. 4. When the signature verification prompt is displayed select to decline the signature.
Expected results:	**For all POS:** 1. The transaction runs successfully from end to end without any errors 2. The POS recognises the transaction as declined 3. A voided EFTPOS receipt **must** be printed **immediately** after the signature has been declined.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 4	**Leave the signature step of the transaction running for 35 minutes**
Rationale:	The POS should wait indefinitely for user input at the signature step.
Test steps:	1. Run a transaction with the visa credit test card 2. When prompted to enter a PIN, don't, put press enter. 3. When offered the choice to accept the signature, wait for 35 minutes 4. Accept the signature
Expected results:	**For all POS:** 1. The transaction runs successfully from end with no errors 2. The POS **does not** implement a timeout at this step. This prompt must be able to run indefinitely as mandated by Paymark. 3. The POS recognises the transaction as accepted 4. A receipt is printed once the transaction reaches the accept with signature step 5. A second receipt is printed once the signature is accepted.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 5	**Attempt to cancel the transaction at the signature stage**
Rationale:	Since the transaction is beyond the financial point of no return, the user must either accept or decline the signature. The POS should not offer a 'third way' to end the transaction.
Test steps:	1. Run a transaction with the Visa Credit test card. 2. Select the credit account 3. Do not enter a PIN when prompted, but press enter 4. When prompted for a signature, try to cancel the transaction. Press keys such as 'C' and 'escape'. Try to cancel the sale, and so forth.
Expected results:	1. It is not possible to cancel the transaction with the only way to proceed is by accepting/declining the signature via the POS prompt.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	The exact steps of this test will vary

Case 6	**Contactless Transaction Under the Floor Limit**
Rationale:	Contactless transactions that are under the floor limit should be able to process immediately without needing to fallback to a contact transaction.
Test steps:	1. Perform a purchase using a Visa PayWave card below the floor limit. 2. Perform a purchase using a MasterCard PayPass card below the floor limit.
Expected results:	**1.** Both transactions complete successfully without the need to fallback to a contact transaction.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 7	**Contactless Transaction Above the Floor Limit**
Rationale:	Contactless transactions that are above the floor limit should prompt for the card to be inserted after it is presented.
Test steps:	1. Perform a purchase using a Visa PayWave card above the floor limit. 2. Insert the card when prompted. 3. Perform a purchase using a MasterCard PayPass card above the floor limit. 4. Insert the card when prompted.
Expected results:	**5.** Both transactions fallback to contact transactions.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 8	EMV Transaction
Rationale:	The POS must be able to handle EMV transactions.
Test steps:	1. Perform a purchase transaction using an EMV card.
Expected results:	2. The transaction is successful.
Deviations from Expected results:	3. For POS that control their own printing the receipt must contain the required EMV data.
Tested by:	
Date:	
Pass / Fail:	
Comments:	

8.1.2 Non-Financial Transactions

Case 9	Perform a manual logon
Rationale:	All POS must have the ability to perform a manual EFTPOS logon.
Test steps:	1. Initiate an EFTPOS logon from within the POS.
Expected results:	**1.** Logon is successful.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

8.1.3 Exception Handling

Case 10	**Reboot the POS during a transaction before the financial point of no return**
Rationale:	If the POS is rebooted during a transaction, there is a risk that the cardholder does not learn the outcome of the transaction.
Test steps:	1. Initiate a transaction 2. At the 'choose account' stage, do a hard reboot of the PC. 3. Start the POS
Expected results:	**For all POS, either:** 1. The interrupted order is restored, the GetLastTransaction method is called and the result returned is used to correctly determine whether to conclude or retender the order. 2. The GetLastTransaction call contains the correct transaction reference **OR** 1. The GetLastTransaction method is called, and the response compared with the POS last recorded transaction. The discrepancy in detail is identified and the POS displays a suitable prompt to the attendant, allowing for manual reconciliation. **For POS that control printing:** 1. No receipt is printed
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 11	**Reboot POS when a transaction is at the signature stage**
Rationale:	If the POS is rebooted during a transaction, there is a risk that the cardholder does not learn the outcome of the transaction. The POS needs to offer the merchant the ability to accept or decline the signature if it crashes at that point.
Test steps:	1. Initiate a transaction with the Visa Credit test card. 2. Choose the credit account. 3. Do not enter a PIN, but press 'enter' when prompted. 4. When asked to accept or decline signature, do a hard reboot of the PC. 5. Start the POS.
Expected results:	1. When the POS has restarted GetLastTransaction() is called and the 'accept with signature' dialogue is displayed **immediately**. There is no opportunity for interaction with the POS prior to this. 2. The attendant must approve or decline the signature before being able to perform any other POS operation. 3. A receipt is printed when the signature is approved or declined. 4. Normal POS operation is then resumed.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 12	**Reboot the POS after a transaction has been accepted but prior to the receipt being printed.**
Rationale:	If the POS is rebooted during a transaction, there is a risk that the cardholder does not learn the outcome of the transaction. The POS needs to be able to resume
Test steps:	1. Initiate a transaction with the Visa Credit test card 2. Choose the credit account 3. Enter PIN when prompted. 4. Hard reboot the PC after the transaction has been accepted but prior to the receipt being printed. 5. Start the POS
Expected results:	1. When the POS has restarted the POS it immediately calls GetLastTransaction(). 2. The POS is updated to recognise that the transaction was successful. OR 3. The POS indicates that the transaction was successful but manual reconciliation is required.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

8.2 TEST CASES THAT APPLY TO POS USING XML INTERFACE

Case 13	**Send an XML message to the POS fragmented over two TCP sends**
Rationale:	To check that the POS implements robust network socket management with the XML listener
Test steps:	1. Add <EnableFragmentMessageTest>1</EnableFragmentMessageTest> to the XML listener configuration. 2. Restart the XML listener 3. Initiate a transaction through the POS using Visa Credit Card, choose credit, and enter PIN 1234. 4. Attempt to complete transaction
Expected results:	1. The transaction completes without any error.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	The XML listener configuration should be returned to normal after this test.

Case 14	**Send two XML messages to the POS within a single XML send**
Rationale:	To check that the POS implements robust network socket management
Test steps:	1. Add <EnablePrefixMessageTest>1</<EnablePrefixMEssageTest> to the XML listener configuration 2. Restart the XML listener service 3. Run a transaction with Visa Credit test card 4. Select the credit account. Enter PIN 1234 and press enter
Expected results:	1. The POS completes the transaction without any errors.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	Restore the XML listener configuration to normal after this test

8.3 TEST CASES THAT APPLY TO ALL POS PROVIDING CUSTOM DIALOGUES

The overall purpose of these test cases is to provide DPS with confidence that the POS is acting as a relay for the messages provided from PX EFTPOS, rather than providing its own messages. In addition to these tests, the tester should monitor the prompts in other test cases for any deviations from those provided by PX EFTPOS.

Case 15	**Cancel a transaction at the account selection stage**
Rationale:	The POS should implement a working button to let the merchant cancel a transaction
Test steps:	1. Initiate a transaction with any test card 2. At the account select screen, click 'cancel' on the POS.
Expected results:	1. The transaction is cancelled successfully
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 16	Run an EOV Transaction
Rationale:	The POS must indicate when the EFTPOS system is running in offline mode
Test steps:	1. Disconnect the network connection on the test machine then attempt to process a transaction with any test card. 2. After the transaction has timed out, wait a further 30 seconds for a reversal to time out. 3. Start a new transaction
Expected results:	1. The POS clearly indicates that EFTPOS is running in offline mode, and the dialogue boxes contain the text “EFTPOS Offline”
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 17	**Run a Manual PAN transaction**
Rationale:	The best practice for running a Manual PAN transaction is to collect the card number through the PINPad.
Test steps:	1. Start a ManPAN transaction and follow it through to completion.
Expected results:	1. There is no way to enter the card number on the POS 2. The POS initiates a ManPan transaction which allows PAN entry on the PINPad 3. The POS must ask for ECi and CSC indicators at the appropriate time.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	This avoids the POS having to gather full credit card details which may result in a security breach.

Case 18	**Attempt to destroy the transaction dialogues**
Rationale:	Transaction dialogues being hidden or destroyed could cause unpredictable outcomes.
Test steps:	1. Try to hide the transaction dialogues. Press alt f4 with the focus on them and control + x and F10 and escape. Try to drag the dialogues behind the POS. Try to minimise them. Try to select them in task manager and kill them. Try everything you can
Expected results:	1. The POS dialogues survive a rigorous attack: they cannot be hidden or destroyed.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	Actual Test steps: will vary.

8.4 TEST CASES THAT APPLY TO POS THAT CONTROL PRINTING

If the POS controls printing, in addition to these cases the tester should pay careful attention to the receipts that are being printed and record any anomalies with the printing.

8.4.1 Test Cases that apply to all POS that control printing

Case 19	**Attempt to run a transaction when the printer is out of paper**
Rationale:	A receipt must be provided for every transaction, but the POS won't be able to provide a receipt if the printer is out of paper.
Test steps:	1. Take the paper out of the printer 2. Try to start a transaction.
Expected results:	1. The POS fails to initiate a transaction. 2. The POS provides sensible messages explaining that transactions cannot be started when the printer is out of paper.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	Steps will vary. The printer must be set up correctly throughout the test.

Case 20	**Attempt a transaction when the printer is turned off**
Rationale:	A receipt must be provided for every transaction, but the POS won't be able to provide a receipt if the printer is switched off
Test steps:	1. Turn off the printer 2. Initiate an EFTPOS transaction.
Expected results:	1. It is not possible to initiate a transaction. 2. The POS provides sensible messages explaining that the transaction cannot be started when the printer is offline
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	Steps will vary. The printer must be set up correctly throughout the test.

Case 21	**Run a signature transaction and check the receipts have been cut correctly**
Rationale:	Incorrect cutting of receipts has in the past been an important source of support calls to DPS.
Test steps:	1. Run a transaction with test card Visa Credit 2. When prompted to enter a PIN, don't, but press enter 3. Click "yes" or "no" when asked to accept the signature
Expected results:	1. Both receipts are correctly cut and look neat and tidy.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 22	**Run a transaction on an EMV card and check the receipt**
Rationale:	EMV cards contain a number of fields on the receipt that magnetic stripes don't. Provided that the POS simply prints the receipt provided by the EFTPOS software, it should have no problem in printing a correct EMV receipt
Test steps:	1. Run a transaction on an EMV chip card.
Expected results:	1. The POS should print a receipt that looks exactly like the receipt in the journal viewer and should contain EMV specific information.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	POS developers should verify this test through code review.

Case 23	Printer Support Review
Test steps:	On the basis of information provided at the beginning of this document, collect printers from your collection that best reflect the deployment of the printers and rerun the cases in this section for each printer.
Expected results:	1. The cases pass for all the printers.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	.

8.4.2 Test cases that apply to all POS controlling printers PX EFTPOS cannot connect to

Case 24	**Reprint a receipt**
Rationale:	If a receipt for a transaction is damaged or illegible, the POS must be able to reprint the receipt. While this functionality can normally be provided by PX EFTPOS, the POS must provide it if it supports printers that PX EFTPOS does not.
Test steps:	1. Run a complete transaction on the visa credit test card 2. Select the credit account 3. When prompted to enter a PIN, don't, but press 'enter' instead. 4. Follow the POS instructions on how to reprint the last receipt
Expected results:	1. The POS reprints the last receipt.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	.

8.5 TIPPING TRANSACTIONS

Case 25	**Tip Auth Approved with Signature**
Rationale:	The POS needs to be able to start tip auth transactions
Test steps:	1. Run a transaction with test card Visa Credit 2. Choose credit account
Expected results:	**For all POS:** **1.** The transaction runs successfully from end to end without any errors **2.** The POS recognises the transaction as approved **3.** The receipt has space on it for entering a TIP **For those POS controlling printing:** **1.** Only one receipt is printed, or two if the POS is printing duplicate receipts.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 26	**When a TIP transaction cannot be initiated, run a purchase transaction with a credit card**
Rationale:	In some situations it will make sense to have all credit card transactions tipping transactions. But a POS' customer may not be set up for tipping at the bank, in which case they might not be able to run credit card transactions at all.
Test steps:	1. Close the POS 2. Set <EnableCreditCardTipping>0</EnableCreditCardTipping> in pxpp_cfg.txt 3. Restart the EFTPOS service 4. Start the POS 5. Run a credit card transaction.
Expected results:	**For all POS:** **1.** The POS initiates a purchase transaction.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 27	**Add a tip of less than 50% of the total value of the pre-auth**
Rationale:	Tips can be added up to 50% of the original bill
Test steps:	1. Run a pre-auth tip that is approved on any test card 2. Use the POS to find the tip transaction 3. Add a TIP of less than 50% of the original amount 4. Check the transaction in the journal viewer
Expected results:	**For all POS:** **1.** The TIP transaction is successfully updated **2.** The POS records the tip correctly
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 28	**Attempt to add a tip of 1c more than 50% the total value of the pre-auth**
Rationale:	Tips can be added up to 50% of the original bill, but not higher. POS should pay attention to the result of Edit Tender in these cases.
Test steps:	1. Run a pre-auth tip that is approved on any test card. The amount must be odd. 2. Use the POS to find the tip transaction 3. Add a TIP of half the original auth amount, rounded up. 4. Check the transaction in the journal viewer
Expected results:	**For all POS:** **1.** The POS displays an intelligible error explaining that tips can only be up to 50% of the original auth **2.** The POS does not record a tip against this transaction
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 29	**Add a tip of exactly 50% of the total value of the pre-auth**
Rationale:	Tips can be added to exactly 50% of a pre-auth. This is a boundary test. POS should pay attention to the result of EditTender in these cases.
Test steps:	1. Run a pre-auth tip transaction with any test card. The amount must be even. 2. Find the transaction in the POS. 3. Add a tip of exactly 50% of the original tip
Expected results:	**For all POS:** **1.** The POS does not display an errors **2.** The POS records the tip against the transaction.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 30	**Add a tip of less than 50%, then another of a higher value but still less than 50%**
Rationale:	To check that the POS remains consistent with PX EFTPOS about what tips have been added to a transaction.
Test steps:	1. Run a pre-auth tip transaction with any test card. 2. Find the transaction in the POS. 3. Add a tip of considerably less than 50% 4. Add a second tip higher than the first tip, but still less than 50% of the original auth.
Expected results:	**For all POS:** **1.** The POS correctly records the first tip **2.** The POS records the second tip as the final tip amount. It does not sum the two tips.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 31	**Add a tip of less than 50%, then another of a lower value**
Rationale:	To check that the POS remains consistent with PX EFTPOS about what tips have been added to a transaction.
Test steps:	1. Run a pre-auth tip transaction with any test card. 2. Find the transaction in the POS. 3. Add a tip of considerably less than 50% 4. Add a second tip lower than the first tip, but still less than 50% of the original auth.
Expected results:	**For all POS:** **1.** The POS correctly records the first tip **2.** The POS updates the tip amount on the transaction to the second amount. It does not sum the two tips.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 32	Void a Tip
Rationale:	To verify that the POS void tip function works. If the POS appears to implement voiding but it does not work correctly, this is a financial integrity issue.
Test steps:	1. Run a tip pre-auth transaction if necessary 2. Find the pre-auth transaction in the POS 3. Void it.
Expected results:	**For all POS:** **1.** You can verify that the TIP is voided by trying to find the TIP in the journal viewer. It should not be there.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 33	**Attempt to update a tip that has been voided**
Rationale:	The POS should not make it appear possible to update voided tests.
Test steps:	1. Run a pre-auth tip transaction with any test card. 2. Find the transaction in the POS. 3. Void the tip 4. Attempt to find the transaction again 5. If you can find the transaction, try to add a tip
Expected results:	**For all POS:** 1. If you can find the transaction, you cannot add a tip.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 34	Upload tips
Rationale:	If the POS implements a button to upload tips, it is an issue of financial integrity that it actually works.
Test steps:	1. Use the POS to upload tips
Expected results:	**For all POS:** 1. The TIP upload processes should start, which includes displaying the total of tips and offering the user the option to upload or not.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

8.6 HOSPITALITY TRANSACTIONS

Case 35	**Pre-auth hospitality transaction approved with signature**
Rationale:	To verify a POS can correctly run a pre-auth hospitality transaction
Test steps:	1. Run a hospitality pre-auth transaction with any visa credit. Choose the credit account.
Expected results:	**For all POS:** 1. You can verify that the transaction is a hospitality transaction in the journal viewer by clicking 'edit tender' and finding it there.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 36	**Update a hospitality transaction to more than 150% of the pre-auth**
Rationale:	Hospitality transactions can be updated to any amount, this is to check that the difference here between tipping and hospitality transactions is respected by the POS.
Test steps:	1. Run a hospitality pre-auth transaction on the visa credit test card. Choose the cheque account. 2. Update it to more than 1.5 times its amount
Expected results:	**For all POS:** 1. The update is successful 2. The hospitality transaction is completed for the amount entered at step 2 3. You can verify these results by checking the journal viewer
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 37	**Update a hospitality transaction to less than 100% of the pre-auth**
Rationale:	To check that hospitality transactions can be completed to less than the amount of the original transaction
Test steps:	1. Run a hospitality pre-auth transaction with any visa credit. Choose the credit account. 2. Update the transaction to less than 100% of the original amount.
Expected results:	**For all POS:** 1. The update is successful 2. The hospitality transaction is updated to the amount specified at step 2. 3. You can verify these results by checking the journal viewer.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 38	**Void a hospitality transaction**
Rationale:	The POS should support voids of hospitality transactions
Test steps:	1. Run a hospitality pre-auth transaction with any visa credit. Choose the credit account. 2. Void the hospitality transaction
Expected results:	**For all POS:** 1. You can verify that the transaction has been voided by checking the journal viewer.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

8.7 CHEQUE TRANSACTIONS

Case 39	**Run a cheque transaction without putting any spaces in the cheque number**
Rationale:	There are numerous sensible ways to enter a cheque number into the POS. The POS should provide the cheque data correctly to PX EFTPOS.
Test steps:	1. Run a cheque transaction with a cheque number that has no spaces in it
Expected results:	**For all POS:** 1. The POS passes the number to PXEFTPOS with the spaces inserted correctly into the cheque number
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	If it is impossible to insert a cheque number like this, that is an acceptable outcome

Case 40	**Run a cheque transaction with the correct spaces in it**
Rationale:	There are numerous sensible ways to enter a cheque number into the POS. The POS should provide the cheque data correctly to PX EFTPOS.
Test steps:	1. Enter a cheque number with spaces in the correct places
Expected results:	**For all POS:** 1. The POS passes the number to PXEFTPOS with the spaces in the correct places
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 41	**Run a cheque transaction with eccentric spaces in the cheque number**
Rationale:	There are numerous sensible ways to enter a cheque number into the POS. The POS should provide the cheque data correctly to PX EFTPOS.
Test steps:	1. Put some spaces in the cheque number at random
Expected results:	**For all POS:** 1. The POS passes the cheque number with the spaces in the right place.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	If it is impossible to insert a cheque number like this, that is an acceptable outcome

8.8 FLYBUYS TRANSACTION

Case 42	**Add a line item**
Rationale:	A POS that supports flybuys transactions needs to be able to add line items that provide information about a transaction
Test steps:	1. Enter a product code, quantity, and amount for a line item 2. Run a flybuys transaction.
Expected results:	**For all POS:** 1. The transaction must be a flybuys transaction 2. The POS line items must be populated correctly
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	DPS will review PX EFTPOS logs to verify this, by checking the <products> tag on the transaction. POS developers should review code. Line items might be added automatically when a sale is created in the POS.

Case 43	**Edit a line item**
Rationale:	A POS that supports flybuys transactions needs to be able to add line items that provide information about a transaction
Test steps:	1. Enter a product code, quantity, and amount for a line item 2. Change the quantity 3. Run a flybuy transaction
Expected results:	**For all POS:** 1. The transaction must be a flybuys transaction 2. The POS line items must be populated correctly
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	DPS will review PX EFTPOS logs to verify this, by checking the <products> tag on the transaction. POS developers should review code. Line items might be added automatically when a sale is created in the POS

Case 44	**Remove a line item**
Rationale:	There are numerous sensible ways to enter a cheque number into the POS. The POS should provide the cheque data correctly to PX EFTPOS in any case.
Test steps:	1. Enter a product code, quantity, and amount for a line item 2. Remove the line item 3. Run a flybuy transaction
Expected results:	**For all POS:** 1. The transaction must be a flybuy transaction 2. The POS passes the cheque number with the spaces in the right place.
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	If it is impossible to insert a cheque number like this, that is an acceptable outcome

8.9 EPAY TRANSACTIONS

<table>
<tr><td>Case 45</td><td>Run Epay Transactions</td></tr>
<tr><td>Rationale:</td><td>Each Epay transaction requires specific set up with an amount and a PAN entered correctly, so each one needs to be run.</td></tr>
<tr><td>Test steps:</td><td>1. Run each an epay transaction for each epay product the POS offers. Fill the results in on the table below
<table>
<tr><th>Product</th><th>Amount</th><th>Product correct</th><th>Amount correct</th><th>Logo Correct</th></tr>
<tr><td>e.g. Vodafone Prepay Topup</td><td>e.g. $20</td><td>e.g. Yes</td><td>e.g. Yes</td><td>e.g. Yes</td></tr>
</table></td></tr>
<tr><td>Expected results:</td><td>For all POS:
1. The POS prints the right logo on the receipt
2. Each transaction is for the right product
3. Each transaction is for the right amount</td></tr>
<tr><td>Deviations from Expected results:</td><td></td></tr>
<tr><td>Tested by:</td><td></td></tr>
<tr><td>Date:</td><td></td></tr>
<tr><td>Pass / Fail:</td><td></td></tr>
<tr><td>Comments:</td><td>This test must be rerun for each Epay product the POS is providing.</td></tr>
</table>

8.10 TEST CASES FOR COMMON OPTIONAL FEATURES

8.10.1 Financial Transactions

Case 46	**Process a refund transaction**
Rationale:	The end of transaction handling should be largely the same for each transaction type, but DPS needs to make sure that the transactions are initiated correctly.
Test steps:	1. Run a refund transaction through the POS, using Visa Credit card 2. Choose the credit account. 3. Do not enter a PIN, just press enter, when prompted 4. Choose 'yes' when offered the choice of accepting or declining the signature
Expected results:	**For all POS:** 1. The transaction runs successfully from end to end without any errors 2. The POS recognises the transaction as approved 3. There are no un-necessary commands such as 'DoReadCard' at the beginning of the transaction 4. 'Refund' is printed on the receipt **For those POS controlling printing:** 1. A receipt is printed for the customer to sign when the 'accept with signature yes/no' prompt appears, and before a choice has been made 2. A second receipt is printed for the customer to take away after a choice has been made
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 47	**Process a cash-out transaction**
Rationale:	The end of transaction handling should be largely the same for each transaction type, but we need to make sure that the transactions are initiated correctly.
Test steps:	1. Run a refund transaction through the POS, using Visa Credit card 2. Choose the credit account. 3. Do not enter a PIN, just press enter, when prompted 4. Choose 'yes' when offered the choice of accepting or declining the signature
Expected results:	**For all POS:** 1. The transaction runs successfully from end to end without any errors 2. The POS recognises the transaction as approved 3. There are no un-necessary commands such as 'DoReadCard' at the beginning of the transaction 4. A line for the amount of the cash out is printed on the receipt, and contains the full amount of the cash out. **For those POS controlling printing:** 1. A receipt is printed for the customer to sign when the 'accept with signature yes/no' prompt appears, and before a choice has been made 2. A second receipt is printed for the customer to take away after a choice has been made
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 48	**Process a purchase with cash transaction**
Rationale:	The end of transaction handling should be largely the same for each transaction type, but we need to make sure that the transactions are initiated correctly.
Test steps:	1. Run a purchase with cash transaction through the POS, using EFTPOScard 2. Choose the cheque / savings account. 3. Do not enter a PIN, just press enter, when prompted 4. Choose 'yes' when offered the choice of accepting or declining the signature
Expected results:	**For all POS:** 1. The transaction runs successfully from end to end without any errors 2. The POS recognises the transaction as approved 3. There are no un-necessary commands such as 'DoReadCard' at the beginning of the transaction 4. The purchase amount on the receipt reflects the amount of the purchase 5. The cash amount on the receipt reflects the amount of the cash out **For those POS controlling printing:** 1. A receipt is printed for the customer to sign when the 'accept with signature yes/no' prompt appears, and before a choice has been made 2. A second receipt is printed for the customer to take away after a choice has been made
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

8.10.2 Non-Financial Transactions

Case 49	Run a settlement
Rationale:	If settlement is supported, DPS needs to check it has been implemented correctly
Test steps:	1. Run an offline transaction 2. Click the settlement button
Expected results:	**For all POS:** 1. There are no errors 2. The POS correctly runs a settlement transaction **For those POS controlling printing:** 1. A receipt is printed for the settlement
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	

Case 50	**POS Correctly processes an enquiry**
Rationale:	If enquiry is supported, DPS needs to confirm it is supported correctly.
Test steps:	1. Run an enquiry through the POS for a recent date with available transactions.
Expected results:	**For all POS:** 1. The POS runs an enquiry 2. The enquiry is for the date selected. **For those POS controlling printing:** 1. A receipt is printed for the enquiry
Deviations from Expected results:	
Tested by:	
Date:	
Pass / Fail:	
Comments:	