****

**ReadyCredit**

**Vantiv Express**

**Functional Specification**

****

****

**Revision 0.06**

**December 13, 2017**

**Table of Contents**

1 Executive Summary 2

1.1 Overview 2

1.2 Contacts 2

1.3 Outstanding Questions 3

2 Project Timeline 4

3 Configuration Changes 5

3.1 Vantiv Express vs. PWS 5

3.2 Vantiv Express Configuration 5

4 API Interfaces 6

4.1 Purchase 6

4.2 Reversal / Void 8

4.3 Sequences of Void/Reversals 10

5 Vantiv Express 11

5.1 Developers Site 11

5.2 Suggested Terminal Settings 11

5.3 SOAP vs. XML Interface 11

6 Use Cases 12

7 PCI 12

8 Testing and Certification 13

8.1 Sonic Router 13

8.2 Test Cards 13

8.3 Test Configurations 13

8.4 Test Spec 14

8.5 Unit Testing 14

8.6 Vantiv Test System 15

8.7 Vantiv Certification 15

9 Appendix 16

9.1 Current Configuration Files 16

9.2 Vantiv PWS Announcement 21

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| REV | DATE | AUTHOR | COMMENTS |
| 1.00 | 12/xx/2017 | Jim Heimbach | Initial Version |

# Executive Summary

## Overview

ReadyCredit needs the NCR kiosk application migrated from Vantiv’s PWS interface to their Express interface.

The PWS implementation will be preserved – allowing ReadyCredit to run both implementations at the same time (on different kiosks). This will allow us to pilot the Express implementation on one kiosk.

## Contacts

### ReadyCard

Jeff Nesheim

Vice President, Operations

Office 612-466-7375

Mobile 612-845-4030

[jeff@readycreditcorp.com](mailto:jeff@readycreditcorp.com)

Rob Holloway

240 753-4815

[Robert.Holloway@readycreditcorp.com](mailto:Robert.Holloway@readycreditcorp.com)

George Hilal

[George@readycreditcorp.com](mailto:George@readycreditcorp.com)

Dan Monette

Network Engineer

Office:612-843-6218

Cell:763-221-3267

[Dan@readycreditcorp.com](mailto:Dan@readycreditcorp.com)

### Vantiv

Kevin Martin, [ETA CPP](https://urldefense.proofpoint.com/v2/url?u=https-3A__www.castleworldwide.com_ETA_registry_RegistrySearch.aspx&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=mnsnWvHsV6ooK9CGS-9WqZi6qYwgSjGyp1W8CHlZQ90&s=TTzSp-FXIDR-MGf8qX9Ntm6E_bZrHCIoGnVlbbYL1f0&e=)

Implementations Consultant

Developer Integrations  
o: 970-335-5126  
[Kevin.Martin2@vantiv.com](mailto:Kevin.Martin2@vantiv.com)

Cari Girdler

815-788-0216

[cari.girdler@vantiv.com](mailto:cari.girdler@vantiv.com)

Doug Bay

[Douglas.bay@vanitv.com](mailto:Douglas.bay@vanitv.com)

## Outstanding Questions

Open questions are highlighted as:

ReadyCard Questions

NCR Questions

KAL Questions

# Project Timeline

Pull together project timeline including Certification process (~~allow 6 to 8 weeks for this~~)

# Configuration Changes

Changes will be required to the configuration files. We will need to preserve the existing values – as this will allow ReadyCredit to pilot/rollout the new Express changes while still utilizing the PWS implementation on some kiosks.

## Vantiv Express vs. PWS

A parameter will be needed to identify which Vantiv interface to use. This will be sent with the local config file:

<!—Use Vantiv Express = TRUE, Vantiv PWS = FALSE -->

<VantivExpress>TRUE</VantivExpress>

## Vantiv Express Configuration

The following will be added to the Comms Config files that are sent from ReadyCredit:

<!-- Vantiv Express connection variables -->

<VantivExpress>

<Account\_ID>1050145</Account\_ID>

<Account\_Token>CB29966D2BDD87…8F9AFBB041201</Account\_Token>

<Application\_ID>8699</Application\_ID>

<Acceptor\_ID>3928907</Acceptor\_ID>

<Application\_Version>????</ Application\_Version >

<Application\_Name>????</ Application\_Name >

<Terminal\_ID>001</Terminal\_ID>

<Express\_XML\_Endpoint>https://certtransaction.elementexpress.com/</ Express\_XML\_Endpoint >

</VantivExpress>

# API Interfaces

We need to document which interfaces we currently use.

* Sequences
* Flows

Reversal sequence

* Approve card
* But don’t dispense gift card due to some issue
* What do we do?

Then map them to the new Express interface.

## Purchase

### PWS Implementation:

**Function:**

Internally, we call authouriseCard() for the purchase transaction [VantivClient object in the .jar file]

We use the following properties to construct the request:

purchaseRequest.setSystemTraceId()

purchaseRequest.setTransactionType()

purchaseRequest.setPaymentType

purchaseRequest.setMerchant()

purchaseRequest.setTransactionTimestamp()

purchaseRequest.setTransactionAmount()

purchaseRequest.setCredit

The same function processes the response; it only verifies if the transaction response contains “settled” or “authorized” then returns true means ok or false means transaction Failed.

**From the log:**

**Vantiv purchase request** - AccountEnding: xxxx3207, Amount: 11.25, SystemTraceId: 1, SequenceNo: 285160

**Vantiv purchase response received** - ReferenceNumber: 490100001, TransactionStatus: not\_approved,

### Express Implementation:

We will post to the HTTP web API.

For Express, the request definition from the documentation

<CreditCardSale xmlns="https://transaction.elementexpress.com">

<Credentials>

<AccountID>1050145</AccountID>

<AccountToken>CB29966D2BDD873CD39E02F7F11018A834EED8826567B3F0BB66C8F90FE8F9AFBB041201</AccountToken>

<AcceptorID>3928907</AcceptorID>

</Credentials>

<Application>

<ApplicationID>8699</ApplicationID>

<ApplicationVersion>1.0</ApplicationVersion>

<ApplicationName>Express.CSharp</ApplicationName>

</Application>

<Terminal>

<TerminalID>01</TerminalID>

<CardholderPresentCode>2</CardholderPresentCode>

<CardInputCode>5</CardInputCode>

<TerminalCapabilityCode>3</TerminalCapabilityCode>

<TerminalEnvironmentCode>2</TerminalEnvironmentCode>

<CardPresentCode>2</CardPresentCode>

<MotoECICode>1</MotoECICode>

<CVVPresenceCode>1</CVVPresenceCode>

</Terminal>

<Card>

<CardNumber>5499990123456781</CardNumber>

<ExpirationMonth>12</ExpirationMonth>

<ExpirationYear>19</ExpirationYear>

</Card>

<Transaction>

<TransactionAmount>6.55</TransactionAmount>

<MarketCode>7</MarketCode>

</Transaction>

</CreditCardSale>

## Reversal / Void

### PWS Implementation:

What function we are calling in PWS: cancelLastAuthourisation()

We use the following properties to construct the request:

cancelRequest.setMerchant()

cancelRequest.setSystemTraceId()

cancelRequest.setSystemTraceId()

cancelRequest.setOriginalSystemTraceId()

cancelRequest.setTransactionType()

cancelRequest.setTransactionTimestamp()

cancelRequest.setOriginalTransactionTimestamp()

cancelRequest.setCancelType()

cancelRequest.setOriginalAmount

cancelRequest.setOriginalSequenceNumber()

cancelRequest.setOriginalReferenceNumber()

cancelRequest.setOriginalAuthCode()

cancelRequest.setOriginalAuthCode()

cancelRequest.setCreditPrimaryAccountNumber()

### Express Implementation:

Reversal Type:

* System reversal (ReversalType=0) is meant to be used as a timeout correction. It accepts inputs of card data and can process without a TransactionID.
  + Use for a timeout or when you receive ExpressResponseCode = 1001 or 1002
* Full reversal (ReversalType=1) is meant to be used as an intentional cancellation of a confirmed, approved transaction.  Typically, a full reversal is triggered by a POS system operator.  It can also be used to reverse Ecomm transactions where you receive mismatched AVS/CVV response codes, which could indicate elevated fraud risk.  Full reversal requires a TransactionID.

Sample:

**<CreditCardReversal** xmlns="<https://transaction.elementexpress.com>"**>**

**<Credentials>**

**<AccountID>**INSERT HERE**</AccountID>**

**<AccountToken>**INSERT HERE**</AccountToken>**

**<AcceptorID>**INSERT HERE**</AcceptorID>**

**</Credentials>**

**<Application>**

**<ApplicationID>**INSERT HERE**</ApplicationID>**

**<ApplicationVersion>**1.0**</ApplicationVersion>**

**<ApplicationName>**Express.CSharp**</ApplicationName>**

**</Application>**

**<Terminal>**

**<TerminalID>**01**</TerminalID>**

**<CardholderPresentCode>**2**</CardholderPresentCode>**

**<CardInputCode>**5**</CardInputCode>**

**<TerminalCapabilityCode>**3**</TerminalCapabilityCode>**

**<TerminalEnvironmentCode>**2**</TerminalEnvironmentCode>**

**<CardPresentCode>**2**</CardPresentCode>**

**<MotoECICode>**1**</MotoECICode>**

**<CVVPresenceCode>**1**</CVVPresenceCode>**

**</Terminal>**

**<Card>**

**<CardNumber>**5499990123456781**</CardNumber>**

**<ExpirationMonth>**12**</ExpirationMonth>**

**<ExpirationYear>**19**</ExpirationYear>**

**</Card>**

**<Transaction>**

**<ReversalType>**1**</ReversalType>** <!--Reversal type of 1 for Full Reversal-->

**<TransactionAmount>**1**</TransactionAmount>**

**<MarketCode>**7**</MarketCode>**

**<TicketNumber>**234436554247**</TicketNumber>**

**<TransactionID>**132465454**</TransactionID>**

**</Transaction>**

**</CreditCardReversal>**

We should document what we will send for both situations (reversal 0 and 1)

In the current application, there is no refund/full reversal 1 option we only do this reversal when the transaction didn’t go through or we didn’t get a response back.

So, we need to do the same with the new API. the only case where we go for a reversal is when we don’t get a confirmation response from the server.

## Sequences of Void/Reversals

There are few cases when we void a transaction:

* if the purchased card was not taken.
* while making a purchase transaction and if we encountered and Exception.
* in idle loop we check if there are any outstanding credit card payments that wasn’t sent.

### Data Storage

*Are we storing any card data after the initial transaction so we can process the reversal/void? We need to understand what data we save to do the reversal/void (for PCI audit).*

We save the card data to a file "C:\Program Files\SVC\Status\CCEvtData.ser" after we send a purchase request, if the transaction was succeeded we remove that data from the file.

### PWS Implementation:

The card object is single and unique only one instance is used in the application, when we save an instance of this object we only exclude the track2Data info all the other values get saved to the file.

private String accountNumber

private String expiryDate

private int orderID

private String amount

private String zip

private String traceRetry

private String txnRefNo

private boolean forcePayment

private boolean voidMessage

private String track1Data

private String track2Data

private String authorizationCode

private String previousAuthorizationCode

private String cardBrand

private String sequenceNumber

private XMLGregorianCalendar timestamp

### Express Implementation:

Need to define any new changes

After sending the initial transaction request, we’ll wait for a response from the server.

In case there was no response we proceed the reversal 0 operation.

In case a response was received no action needed.

System reversal (ReversalType=0) is meant to be used as a timeout correction, exactly as Jim described in the first e-mail this morning. It accepts inputs of card data and can process without a TransactionID.

Full reversal (ReversalType=1) is meant to be used as an intentional cancellation of a confirmed, approved transaction. Typically a full reversal is triggered by a POS system operator. It can also be used to reverse Ecomm transactions where you receive mismatched AVS/CVV response codes, which could indicate elevated fraud risk. Full reversal requires a TransactionID.

Waiting to hear from Kevin in regards to PCI compliance on reversal types of 0

System Reversals require that the card data be submitted in the same format it was originally submitted, but that card data must always be included on a System Reversal. As far as storing it while waiting for a timeout, they definitely don't want to write it to disk. Our recommendation is to store the transaction data in memory within their application for the few seconds while the transaction is in progress. If the transaction is successful, that data can be wiped from memory, but if there's a communication error or timeout, then the same card data kept in memory for that transaction can be used for the System Reversal.

# Vantiv Express

## Developers Site

From Vantiv: If not already done, please register as a user on the [Vantiv ONE developer site](https://urldefense.proofpoint.com/v2/url?u=https-3A__developer.vantiv.com_&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=mnsnWvHsV6ooK9CGS-9WqZi6qYwgSjGyp1W8CHlZQ90&s=nAcmq5ov7BZV05M02QanVGs9BZ_0m1bPLWqPZz0cXfo&e=).  This will get you access to our API documentation.  Once registered, visit this link for the primary API info: [Express Interface API v2.8.4](https://urldefense.proofpoint.com/v2/url?u=https-3A__developer.vantiv.com_docs_DOC-2D1353&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=mnsnWvHsV6ooK9CGS-9WqZi6qYwgSjGyp1W8CHlZQ90&s=vi9kGtJ0H1ZfuupJ7DNpyPTRZpfrd9xUXzLsPDchLDQ&e=)

## Suggested Terminal Settings

Also, see Express Suggested Terminal Settings (separate file).  These settings are required for certification and are the most common hurdle that slows down our certifications.  Please use the terminal settings from page 3, “Retail Swiped”, for Sale transactions.  For Reversals, use the same terminal settings with one exception, **CardInputCode=4** (Manual Keyed).

## SOAP vs. XML Interface

The Express platform supports both SOAP and XML interfaces.  However, the SOAP interface is no longer being updated and will ultimately fall behind the technology curve.  Vantiv recommends using the XML interface if possible.

# Use Cases

Need to document the different use cases and sequences

* Credit Approval / CARD Loaded successfully
* Credit Approval / CARD not loaded successfully / Reversal Credit Auth
* Credit Approval / network down / queue Credit Reversal (I think we have a file that holds this info)
* Credit Approval sent but no response / send reversal
* Credit Reversal sent but no response /

# PCI

What are we storing?

How do we handle reversals (just store ID, sequence ??)

# Testing and Certification

## Sonic Router

KAL will need to connect to their sonic wall router (need to figure out where this is Mexico or Cincinnati).  Also, ReadyCredit will need to reprogram their sonic wall (they did Stuber’s a while back).

Need to get Heimbach and KAL’s router reprogrammed

## Test Cards

Joe Stuber received the following from Vantiv:

2 Sets of 10 Test Cards Received.

The 10 Test Cards are labeled

1. Test card #1
2. Test card #2
3. Test card #3
4. Test card #4
5. Test card #5
6. Test card #6
7. Test card #7
8. Vantiv Debit Test Card
9. Test GIFT CARD 5% Cash Back
10. Test GIFT CARD 0% Cash Back

## Test Configurations

Email exchange that can be referenced when setting up test kiosk:

Jeff,

Are you OK if we connect and use some of the 999XX ids in the next couple of weeks to test the Receipt XML change?

Do you still ignore transactions with 999XX that come across?

Joe.

Joe,

Yes, no problem using the 999xx ids.  That is correct, the 999xx transactions do not go to MetaBank cardholder funding account.  We consider all 999xx transactions as test transactions.

Jeff Nesheim

Jeff,

Trying to connect via 99910 kiosk getting connection error.

What are should I be using in terminal config for the Comms Config URL and Download Server URL?

Joe

Joe,

When we switched data centers most recently we switched to FQDN for these. It will simplify future moves if the need should arise. Please see below.

[http://config.readycreditcorp.com/readycreditv5/commsconfig.aspx](https://urldefense.proofpoint.com/v2/url?u=http-3A__config.readycreditcorp.com_readycreditv5_commsconfig.aspx&d=DwMFAg&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=alXE51ZBYeAXMcGX1IaiE7lzYTcwQgJ_frxwedHbCaI&s=IGs2kCA5r9Sk0v7BgAWh_FfvGym2WHI5kOifI22pczc&e=)

[http://version.readycreditcorp.com/KioskShare/R8PROD/MasterSM.xml](https://urldefense.proofpoint.com/v2/url?u=http-3A__version.readycreditcorp.com_KioskShare_R8PROD_MasterSM.xml&d=DwMFAg&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=alXE51ZBYeAXMcGX1IaiE7lzYTcwQgJ_frxwedHbCaI&s=LD1evRBGoBxZMupgeUy2ziqDvLlIuPHtqG_PIjlght4&e=)

Dan Monette

Dan,

The below change along with changing the DNS entries to:

10.61.114.11

10.61.114.12

worked and I am able to come up on 99910 kiosk ID.

Joe

## Test Spec

## Unit Testing

## Vantiv Test System

Jim Heimbach registered for a test environment on 11/14/2017. Emailed details:

To get started, create an [Express Test Account](https://urldefense.proofpoint.com/v2/url?u=http-3A__www.elementps.com_Create-2Da-2DTest-2DAccount&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=mnsnWvHsV6ooK9CGS-9WqZi6qYwgSjGyp1W8CHlZQ90&s=YfdGhmgd1ebadRPv4MqH4JkrwLNvKX8XuLVFMbMPnD8&e=).  This step provides your sandbox credentials and endpoint URL’s, as well as your ApplicationID which will remain constant across all your merchants using the new solution.

**Express Test Credentials**  
  
Dear James Heimbach:  
  
Your new test account on Express has been set up successfully! **Please save the Test Credentials below in your records as they are needed to submit test transactions. Express Test Credentials.**   
  
AccountID: **1050145**  
AccountToken: **CB29966D2BDD873CD39E02F7F11018A834EED8826567B3F0BB66C8F90FE8F9AFBB041201**  
ApplicationID: **8699**  
AcceptorID: **3928907**  
  
**Express Test URL's:**  
[XML (https://certtransaction.elementexpress.com)](https://urldefense.proofpoint.com/v2/url?u=https-3A__certtransaction.elementexpress.com&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=Bhgn_fneuOqjKCXYWPTwaOgjfrUOoDV4lIc8_DhKBak&s=7EVtOh3S__wDlhtdWC9Ud-7PQ8xh2lMwLBuLCjjua34&e=)  
[SOAP (https://certtransaction.elementexpress.com/express.asmx)](https://urldefense.proofpoint.com/v2/url?u=https-3A__certtransaction.elementexpress.com_express.asmx&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=Bhgn_fneuOqjKCXYWPTwaOgjfrUOoDV4lIc8_DhKBak&s=sIZl2YT1_0e_BEUSowhRfideclbhaDmVnV4vK7BcWFU&e=)  
[Hosted Payments (https://certtransaction.hostedpayments.com)](https://urldefense.proofpoint.com/v2/url?u=https-3A__certtransaction.hostedpayments.com&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=Bhgn_fneuOqjKCXYWPTwaOgjfrUOoDV4lIc8_DhKBak&s=y0F0rd8X-9wK5qjWmTXXl62GDbp7zLU1aKifBoBiGIY&e=)  
  
**Important Documents:**   
[Express Interface Specification:](https://urldefense.proofpoint.com/v2/url?u=https-3A__www.vantiv.com_content_dam_vantiv_developers_Express-2DInterface-2DSpecification.pdf-3FlogActivity-3Dtrue&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=Bhgn_fneuOqjKCXYWPTwaOgjfrUOoDV4lIc8_DhKBak&s=uhG1Lt5hLiQ326FGpNlWPeD9tZF9YLaVQBJwEYYXkTg&e=) - Technical Document for Solution Providers.  
  
**Certification Documents:**   
[Express Certification Process](https://urldefense.proofpoint.com/v2/url?u=https-3A__www.vantiv.com_expresscertificationoverview&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=Bhgn_fneuOqjKCXYWPTwaOgjfrUOoDV4lIc8_DhKBak&s=6D6Y5HEUbhvTk1iUUuMleH_ZIHU9GWzAcE4IaMHyrYQ&e=) - A high-level overview of getting your application certified on the Vantiv Express platform.  
[Certification Details](https://urldefense.proofpoint.com/v2/url?u=https-3A__www.vantiv.com_expresscertificationdetails&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=Bhgn_fneuOqjKCXYWPTwaOgjfrUOoDV4lIc8_DhKBak&s=36iB87KCVD5UVzM7FGK2JDxOIjlNG_8QvtmZLvuSmVs&e=) - Important information you should know prior to starting your integration.  
[Processing Scenarios](https://urldefense.proofpoint.com/v2/url?u=https-3A__www.vantiv.com_expressprocessingscenarios&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=Bhgn_fneuOqjKCXYWPTwaOgjfrUOoDV4lIc8_DhKBak&s=KdRdr41inQPN2_Eqng-CRjN0U_8ehCsjb_DBZoFs608&e=)- Identifying the transaction scenarios your application supports (i.e. Retail, Direct Marketing, Ecommerce, etc.).  
[Request for Certification](https://urldefense.proofpoint.com/v2/url?u=https-3A__www.vantiv.com_requestforcertification&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=Bhgn_fneuOqjKCXYWPTwaOgjfrUOoDV4lIc8_DhKBak&s=UjMQsaoVabU2zexuuddHN1of-dZTNaIrOhIyss0S9dk&e=)- Starts the certification process.  
[Receipt Requirements](https://urldefense.proofpoint.com/v2/url?u=https-3A__developer.vantiv.com_docs_DOC-2D2135&d=DwMGaQ&c=gJN2jf8AyP5Q6Np0yWY19w&r=XZyqNIVd__Dd3Z1X7FbQI7xVWFEPQQtvfsOHU75sM7Y&m=Bhgn_fneuOqjKCXYWPTwaOgjfrUOoDV4lIc8_DhKBak&s=Sr9lBE0oTfj4jexSqqsuB8Xb5q79uuYE5qp_tIEllMY&e=)- Note that EMV integrations also require the ability to provide EMV decline receipts to cardholders.  
  
Please feel free to contact us by e-mail at [Certification@vantiv.com](mailto:Certification@vantiv.com), or simply by replying to this e-mail, with any questions you may have.

## Vantiv Certification

Need to document this process. Tasks, time, etc.

# Appendix

## Current Configuration Files

Jeff provided the following files:

* 00096 is a Visa site
* 00097 is a Mastercard site

### Visa Config Files

#### Local Config File - Visa

<response>

<getConfigResponse>

<status>S</status>

<kiosk>

<id></id>

<date></date>

<time></time>

</kiosk>

<error>

<errorNumber>0</errorNumber>

<errorMessage></errorMessage>

</error>

<!-- Branding tag specifies dir to copy custom files from... -->

<Branding>VisaGift</Branding>

<!-- Cash Box Limit to go OOS and Denominations accepted -->

<cashBoxLimit>2000000</cashBoxLimit>

<acceptedBillDenominations>1,5,10,20,50,100</acceptedBillDenominations>

<!-- Bin used for Purchase new card. Min/Max/Fee and Display Activation screen option -->

<PurchaseBIN>451002</PurchaseBIN>

<purchaseCardMax>50000</purchaseCardMax>

<purchaseCardMin>1000</purchaseCardMin>

<purchaseCardFee>0</purchaseCardFee>

<activationNotice>No</activationNotice>

<!-- Communications variables -->

<commsMaxRetries>3</commsMaxRetries>

<commsMaxTime>120000</commsMaxTime>

<!-- Receipt Values -->

<Receipt-Location1>Ready Credit Corporation</Receipt-Location1>

<Receipt-Location2>1004 NW 65 Ave</Receipt-Location2>

<Receipt-Location3>Miami, FL 33126</Receipt-Location3>

<Receipt-Location4>(866) 331-1258</Receipt-Location4>

<merchantLicense></merchantLicense>

<!-- Used to Override Reload Fees at this kiosk for the 2 BINS listed -->

<OverrideReloadFees>FALSE</OverrideReloadFees>

<OverrideReloadBIN1>455915</OverrideReloadBIN1>

<OverrideReloadBIN2>455918</OverrideReloadBIN2>

<OverrideReloadFeePermanentCard>200</OverrideReloadFeePermanentCard>

<OverrideReloadFeeInstantCard>200</OverrideReloadFeeInstantCard>

<!-- Only segments listed will show in attract loop sequence -->

<AttractLoopPlayList>1,2,5,6</AttractLoopPlayList>

<!-- TRANSIT Variables, only used when AllowTransit = TRUE -->

<AllowTransit>false</AllowTransit>

<maxPassLimit>5</maxPassLimit>

<passesList>

<pass name="Day" salePeriodType="0" salePeriod="Always" price="500" descriptionEnglish="Day Pass (good for today or any future day)" descriptionSpanish=""></pass>

<pass name="Weekly" salePeriodType="1" salePeriod="Always" price="2500" descriptionEnglish="Weekly Pass (good for 7 days once activated)" descriptionSpanish=""></pass>

<pass name="Monthly" salePeriodType="2" salePeriod="Always" price="7500" descriptionEnglish="Monthly Pass (good for 30 days once activated)" descriptionSpanish=""></pass>

</passesList>

<!-- The kiosk will go OOS at SleepTime and into Service at StartTime, Must be >10 minutes apart -->

<StartTime>0600</StartTime>

<SleepTime>2300</SleepTime>

<SleepOOSMsg><![CDATA[Hours Of Operation <br>This ReadySTATION is Currently Closed. <br>The hours of operation are 6:00AM to 11:00PM <br> <br> Horas de operacion <br>Esta ReadySTATION esta cerrada. <br> Las horas de operacion son de 6:00AM a 11:00PM ]]></SleepOOSMsg>

<!-- Credit Card as Payment Variables, only used when AllowCreditPayment = TRUE -->

<AllowCreditPayment>TRUE</AllowCreditPayment>

<CreditButtonText>Visa</CreditButtonText>

<CreditButtonTextSpanish>Visa\*</CreditButtonTextSpanish>

<MinCreditPaymentAmount>1000</MinCreditPaymentAmount>

</getConfigResponse>

</response>

#### Comms Config File - Visa

<response>

<getCommsConfigResponse>

<status>S</status>

<kiosk>

<id></id>

<date></date>

<time></time>

</kiosk>

<error>

<errorNumber>0</errorNumber>

<errorMessage></errorMessage>

</error>

<ConnectGrp>

<Connection>L</Connection><!-- M=Modem,L=LAN(DSL, Broadband,etc)-->

<ConnectionSpeed>56</ConnectionSpeed><!-- KB connection speed -->

<PingURL> $APIPATHKioskPing.aspx </PingURL>

<PingTimeout>30000</PingTimeout>

</ConnectGrp>

<GalileoGrp>

<Provider\_ID>249</Provider\_ID>

<Provider\_UID>9EEgYv-0020</Provider\_UID>

<Provider\_PWD>\*\*REMOVED\*\*</Provider\_PWD>

<Product\_ID>1037</Product\_ID>

<Location\_ID>00096</Location\_ID>

<Location\_Type>2</Location\_Type>

<LoadType></LoadType>

<ServerURL>https://api-ncr.pd.gpsrv.com/intserv/4.0/</ServerURL>

<StartingMessageID>0000000</StartingMessageID>

</GalileoGrp>

<!-- i2c Card Processor Connection Values -->

<i2cGrp>

<AcquirerId>RccCardacq</AcquirerId>

<UserID>Rcci2cP</UserID>

<Password>RccI2cPrOd</Password>

<WebServerURL>https://ws2.mycardplace.com:6443/MCPWebServiceDL/services/MCPWSHandler</WebServerURL>

</i2cGrp>

<!-- Vantiv connection variables -->

<VantivGrp>

<Merchant\_ID>4445013677203</Merchant\_ID>

<Merchant\_Name>VISA-MIAMI#96</Merchant\_Name>

<Terminal\_ID>001</Terminal\_ID>

<Network\_Routing>E7</Network\_Routing>

<User\_Name>s.MID20.PAY.WS.PROD</User\_Name>

<Password>\*\*REMOVED\*\*</Password>

<PrimaryServerURL>https://ws.vantiv.com/merchant/payments/v5</PrimaryServerURL>

</VantivGrp>

<SupervisorServerGrp>

<SVFunctionsURL>$APIPATHsuperfunc.aspx</SVFunctionsURL>

<EndOfDayURL>$APIPATHeod.aspx</EndOfDayURL>

<PostTranURL>$APIPATHposttran.aspx</PostTranURL>

<getConfigURL>$APIPATHgetconfig.aspx</getConfigURL>

<getSupervisorInfoURL>$APIPATHgetsupervisorinfo.aspx</getSupervisorInfoURL>

<ApplyReadyCARDURL>$APIPATHverifyandcreatecard.aspx</ApplyReadyCARDURL>

</SupervisorServerGrp>

</getCommsConfigResponse>

</response>

### Mastercard Config Files

#### Local Config File - Mastercard

<response>

<getConfigResponse>

<status>S</status>

<kiosk>

<id></id>

<date></date>

<time></time>

</kiosk>

<error>

<errorNumber>0</errorNumber>

<errorMessage></errorMessage>

</error>

<!-- Branding tag specifies dir to copy custom files from... -->

<Branding>MCGift</Branding>

<!-- Cash Box Limit to go OOS and Denominations accepted -->

<cashBoxLimit>2000000</cashBoxLimit>

<acceptedBillDenominations>5,10,20,50,100</acceptedBillDenominations>

<!-- Bin used for Purchase new card. Min/Max/Fee and Display Activation screen option -->

<PurchaseBIN>51285702</PurchaseBIN>

<purchaseCardMax>50000</purchaseCardMax>

<purchaseCardMin>2000</purchaseCardMin>

<purchaseCardFee>0</purchaseCardFee>

<activationNotice>No</activationNotice>

<!-- Communications variables -->

<commsMaxRetries>3</commsMaxRetries>

<commsMaxTime>120000</commsMaxTime>

<!-- Receipt Values -->

<Receipt-Location1>Ready Credit Corporation</Receipt-Location1>

<Receipt-Location2>5555 Winghaven Blvd</Receipt-Location2>

<Receipt-Location3>O'Fallon, MO 63368</Receipt-Location3>

<Receipt-Location4>(866) 345-2315</Receipt-Location4>

<merchantLicense></merchantLicense>

<!-- Used to Override Reload Fees at this kiosk for the 2 BINS listed -->

<OverrideReloadFees>FALSE</OverrideReloadFees>

<OverrideReloadBIN1>455915</OverrideReloadBIN1>

<OverrideReloadBIN2>455918</OverrideReloadBIN2>

<OverrideReloadFeePermanentCard>200</OverrideReloadFeePermanentCard>

<OverrideReloadFeeInstantCard>200</OverrideReloadFeeInstantCard>

<!-- Only segments listed will show in attract loop sequence -->

<AttractLoopPlayList>1,2,5,6</AttractLoopPlayList>

<!-- TRANSIT Variables, only used when AllowTransit = TRUE -->

<AllowTransit>false</AllowTransit>

<maxPassLimit>5</maxPassLimit>

<passesList>

<pass name="Day" salePeriodType="0" salePeriod="Always" price="500" descriptionEnglish="Day Pass (good for today or any future day)" descriptionSpanish=""></pass>

<pass name="Weekly" salePeriodType="1" salePeriod="Always" price="2500" descriptionEnglish="Weekly Pass (good for 7 days once activated)" descriptionSpanish=""></pass>

<pass name="Monthly" salePeriodType="2" salePeriod="Always" price="7500" descriptionEnglish="Monthly Pass (good for 30 days once activated)" descriptionSpanish=""></pass>

</passesList>

<!-- The kiosk will go OOS at SleepTime and into Service at StartTime, Must be >10 minutes apart -->

<StartTime>0300</StartTime>

<SleepTime>2300</SleepTime>

<SleepOOSMsg><![CDATA[Hours Of Operation <br>This ReadySTATION is Currently Closed. <br>The hours of operation are 3:00AM to 11:00PM <br> <br> Horas de operacion <br>Esta ReadySTATION esta cerrada. <br> Las horas de operacion son de 3:00AM a 11:00PM ]]></SleepOOSMsg>

<!-- Credit Card as Payment Variables, only used when AllowCreditPayment = TRUE -->

<AllowCreditPayment>TRUE</AllowCreditPayment>

<CreditButtonText>MasterCard</CreditButtonText>

<CreditButtonTextSpanish>MasterCard\*</CreditButtonTextSpanish>

<MinCreditPaymentAmount>1000</MinCreditPaymentAmount>

<!-- i2c Variables -->

<CashTransactionTypeId></CashTransactionTypeId>

<CreditTransactionTypeId>GM</CreditTransactionTypeId>

<PurchaseCardDescription>MasterCard Gift Card Purchase</PurchaseCardDescription>

<ReloadCardDescription>ReadySTATION Reload</ReloadCardDescription>

<NameAndLocation>MC-O'FALLON#97</NameAndLocation>

<PurchaseApplyFee>N</PurchaseApplyFee>

<ReloadApplyFee>N</ReloadApplyFee>

<BalanceApplyFee>N</BalanceApplyFee>

</getConfigResponse>

</response>

#### Comms Config File - Mastercard

<response>

<getCommsConfigResponse>

<status>S</status>

<kiosk>

<id></id>

<date></date>

<time></time>

</kiosk>

<error>

<errorNumber>0</errorNumber>

<errorMessage></errorMessage>

</error>

<ConnectGrp>

<Connection>L</Connection><!-- M=Modem,L=LAN(DSL, Broadband,etc)-->

<ConnectionSpeed>56</ConnectionSpeed><!-- KB connection speed -->

<PingURL> $APIPATHKioskPing.aspx </PingURL>

<PingTimeout>30000</PingTimeout>

</ConnectGrp>

<GalileoGrp>

<Provider\_ID>249</Provider\_ID>

<Provider\_UID>9EEgYv-0020</Provider\_UID>

<Provider\_PWD>\*\*REMOVED\*\*</Provider\_PWD>

<Product\_ID>1272</Product\_ID>

<Location\_ID>00097</Location\_ID>

<Location\_Type>2</Location\_Type>

<LoadType></LoadType>

<ServerURL>https://api-ncr.pd.gpsrv.com/intserv/4.0/</ServerURL>

<StartingMessageID>0000000</StartingMessageID>

</GalileoGrp>

<!-- Vantiv connection variables -->

<VantivGrp>

<Merchant\_ID>4445013677153</Merchant\_ID>

<Merchant\_Name>MC-O'FALLON#97</Merchant\_Name>

<Terminal\_ID>001</Terminal\_ID>

<Network\_Routing>E7</Network\_Routing>

<User\_Name>s.MID20.PAY.WS.PROD</User\_Name>

<Password>\*\*REMOVED\*\*</Password>

<PrimaryServerURL>https://ws.vantiv.com/merchant/payments/v5</PrimaryServerURL>

</VantivGrp>

<!-- i2c Card Processor Connection Values -->

<i2cGrp>

<AcquirerId>RccCardacq</AcquirerId>

<UserID>Rcci2cP</UserID>

<Password>RccI2cPrOd</Password>

<WebServerURL>https://ws2.mycardplace.com:6443/MCPWebServiceDL/services/MCPWSHandler</WebServerURL>

</i2cGrp>

<SupervisorServerGrp>

<SVFunctionsURL>$APIPATHsuperfunc.aspx</SVFunctionsURL>

<EndOfDayURL>$APIPATHeod.aspx</EndOfDayURL>

<PostTranURL>$APIPATHposttran.aspx</PostTranURL>

<getConfigURL>$APIPATHgetconfig.aspx</getConfigURL>

<getSupervisorInfoURL>$APIPATHgetsupervisorinfo.aspx</getSupervisorInfoURL>

<ApplyReadyCARDURL>$APIPATHverifyandcreatecard.aspx</ApplyReadyCARDURL>

</SupervisorServerGrp>

</getCommsConfigResponse>

</response>

## Vantiv PWS Announcement

**PWS End of Service – FAQ** June 26, 2017

1) When will PWS be decommissioned?

* 10 PM MST / 12 AM EST on Thursday June 7, 2018

2) Why is PWS being decommissioned?

* Vantiv has acquired several assets related to transaction processing over the past few years. During our strategic planning process, we determined there are platforms that can more effectively provide the services that PWS currently provides. The other platform options provide more features and have proven to be more reliable and scalable. For those reasons, it is in the best interest of our clients to migrate from PWS to a more sustainable platform.

3) What happens between now and June 7, 2018?

* All existing PWS customers will be contacted to discuss migration to other Vantiv platforms. It’s Vantiv’s goal to introduce alternative platforms that not only support existing needs but also expose additional features and tools not available on PWS.
* Notifications and information will continue to be distributed to all existing customers throughout the process.

4) What happens after June 7, 2018?

* PWS will no longer be accessible including endpoints used for transaction processing.

5) PWS will continue to be in maintenance mode until June 7, 2018. What does maintenance mode mean?

* No new functionality will be added to the platform and only critical defect repairs and patches will be implemented.
* Support will still be available to help with any issues you may encounter.

6) Can I add any additional features?

* No new or re-certifications of existing applications are supported.
* If you need a new feature, such as Apple Pay, Android Pay or EMV, please reach out to your Vantiv Relationship Manager to discuss what options work best for your solution.

7) What will happen with the transaction data stored in PWS?

* Vantiv maintains records of transactions processed on the PWS platform according to financial regulatory rules.
* Transaction data will not be available for retrieval through PWS after June 7, 2018 and will be purged according to Vantiv’s standard data retention policy. Please note that tools like Vantiv IQ will continue to provide access to your processing data.

8) How will my tokenized data be managed?

* Omni-Token will work in PWS as well as target platforms like LitleXML, Express, 610 and ISO8583.
* Vantiv can assist with the migration and re-tokenization of external sensitive card data. If this is something you would be interested in, please reach out to your Vantiv Relationship Manager no later than December 31, 2017 to allow ample time for this process to take place.

9) What will happen with the transaction data stored in PWS?

* PWS is a stateless routing engine to Vantiv’s 610 and ISO8583 message format. As such, data is not stored in PWS but rather the authorization platform RAFT.
* It is very important that you perform all reversals (void) and close the batch before transitioning transaction processing over to your new solution. Transactions such as returns are considered stand alone and as such will be available in the new API for processing. If you are using omni-token, you will be able to use the token to initiate the return as well as card data.
* Vantiv maintains records of transactions processed on the PWS platform according to financial regulatory rules.
* Since PWS is a pass through to the ISO8583 format, tools like Vantiv IQ, the eMAF file and the CHARTS file will continue to provide access to your processing data.

10) What will happen with my merchant accounts?

* PWS is a pass through interface that routes transactions to the 610 or RAFT platforms. End of Service for PWS will have no impact on the merchant accounts you are using to process transactions. During your migration efforts, your merchant accounts will be setup to process on the target platforms you choose.

11) Will the Sandbox continue to be available?

* No, the Sandbox and Production environments of the platform will both be decommissioned on June 7, 2018.

12) Will I still have access to Direct or Vantiv IQ?

* Yes, Direct and Vantiv IQ will continue to be available.

13) I need EMV, what are my options?

* Vantiv provides various types of EMV solutions; from semi integrated to fully customizable workflows across different platforms. This should be discussed with your Vantiv representative as you identify your new go forward platform.

14) Who should I reach out to if I have questions? Are people still actively supporting PWS?

* There is still active support. Please contact your Vantiv Relationship Manager. If you do not have an assigned Relation

\*\*\* END OF DOCUMENT \*\*\*