|  |
| --- |
|  |
| System Design Specificaiton - NCB and eMcrey for implementation of Cortex.  Version 1.5 August 2015 |



[Revision History 3](#_Toc427687989)

[Introduction and overview 4](#_Toc427687990)

[Document Purpose 4](#_Toc427687991)

[Audience 4](#_Toc427687992)

[Reference Document 4](#_Toc427687993)

[1. Cortex Architecture 5](#_Toc427687994)

[Cortex Core 5](#_Toc427687995)

[Technologies 7](#_Toc427687996)

[2.Services. 8](#_Toc427687997)

[2.1 List of Web Services and what field will be used for routing : 8](#_Toc427687998)

[2.2 Predictive dialler new services 12](#_Toc427687999)

[2. Batch Files. 35](#_Toc427688000)

[3.1 Complete list of Batch Files 35](#_Toc427688001)

[3.2 Batch Files handling during Co-existence of VISION+ and Cortex 38](#_Toc427688002)

[3.2.1 Out Batch File Processing 38](#_Toc427688003)

[3.2.2 IN Batch File Processing 39](#_Toc427688004)

[3.3 Smart Hub 40](#_Toc427688005)

[3.4 Identify systems that will use the new BINS 41](#_Toc427688006)

[3.5 Control M Changes 42](#_Toc427688007)

[ 42](#_Toc427688008)

[3.6 Cordys 43](#_Toc427688009)

[3.6 Middleware 44](#_Toc427688010)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Revision Notes | Author |
| 16/06/2015 | 1.0 | Initial draft | eMCrey |
| 19/06/2015 | 1.1 | Updated after meeting with NCB IT Management team | eMcrey |
| 21/6/2015 | 1.2 | Updated with Cortex Architecture | eMcrey |
| 27/7/2015 | 1.3 | Updated SmartHub, Arch diagram and Predictive dialer services | eMcrey |
| 9th August | 1.4 | Updated with wsdl’s for Predictive dialler | eMcrey |
| 18th August | 1.5 | Updated consolidate file and BIN config infromation & Control M design | Junaid/eMcrey |

# Introduction and overview

## Document Purpose

NCB aims to move the credit card issuance, authorization and all related credit card activities in-house. There will be a co-existence stage with Vision+ and Cortex and this document aims to clarify exactly what development needs to take place within NCB and additional programs to be written by eMCrey in order to support both the Vision+ system as well as Cortex during this phase of the project.

## Audience

The document is intended for NCB and eMCrey management and development teams in order to plan and resource accordingly.

## Reference Document

System requirement specification - SRS V1.3 point of reference.

# Cortex Architecture

# Cortex Core

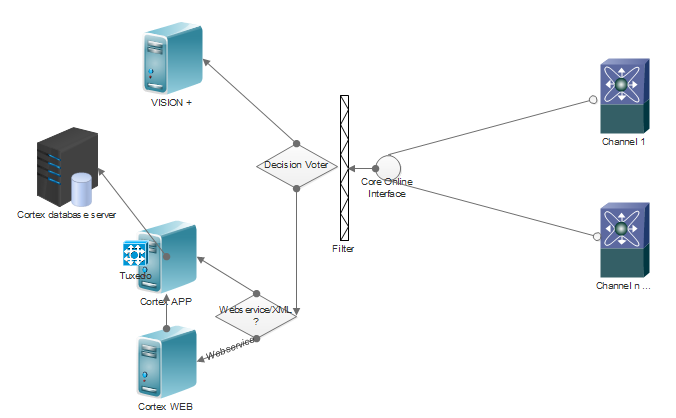


Figure - Flow

One of the key themes of the Cortex core is inversion of control. In its broadest sense, this means that the framework handles responsibilities on behalf of the components that are managed within its context. The components themselves are simplified since they are relieved of those responsibilities. In each case, the end result is a system that is easier to test, understand, maintain, and extend.

One of the technologies used for online interfaces is the **Spring Integration**. It is motivated by these same goals and principles. It extends the spring programming model into the messaging domain and builds upon spring’s existing enterprise integration support to provide an even higher level of abstraction. It supports message-driven architectures where inversion of control applies to runtime concerns, such as when certain business logic should execute and where the response should be sent. It supports routing and transformation of messages so that different transports and different data formats can be integrated without impacting testability. In other words, the messaging and integration concerns are handled by the framework, so business components are further isolated from the infrastructure and developers are relieved of complex integration responsibilities.

# Technologies

The technologies used are all based on JRE7+,

* JRE 7+
* Spring Integration
* Apache thrift
* Axis – webservices
* Spring framework
* Spring batch framework
* Spring decision voter
* Cortex TCP IP – *ISO Custom*
* Sencha framework
* Oracle Tuexdo

As visualized in the above image, the channels shall hit the cortex core online interface, the decision voter and based on specific configurable flag, will decide whether the request hits VISION+ or cortex.

If it hits Cortex, another decision voter will have to decide if the call is designated to reach webservices or cortex app interfaces.

If it is a webservice call, it shall go to Cortex WEB which will fetch the business logic based by contacting the cortex core on app server.

# 2.Services.

All Middleware messages from the day Cortex goes live will be routed either to Vision+ or to Cortex. The below section describes which field will be used to interpret which system the message will be routed to.

## List of Web Services and what field will be used for routing :

|  |  |
| --- | --- |
| Service Name | Routing Field |
| BillPayment / BillPaymentSadad / CreditCardAuthorize / CreditCardTransferCash / SadadBillPayment / SadadFeePayment / SawaRecharge | CreditCardNo |
| CashBack Related (CashBackEarned and CashBackTransactions) | CreditCardNo |
| CreditCardNo |
| CreditCardNo |
| CreditCardNo |
| CreditCardNo |
| CreditCardNo |
| CreditCardActivate | CreditCardNo |
| CreditCardApplicationDetail | AppID (prefix can be used for Cortex related applications and based on which the routing can be done) |
| CreditCardApplicationInquiryDetail | UniqueId/ApplicationNo |
| CreditCardsApplications | SearchFor |
| CreditCardApplicationSearch | Value |
| CreditCardBalance | CreditCardNo |
| CreditCardDetails | CreditCardNo |
| CreditCardBlock | CreditCardNo |
| CreditCardChangeAddress | CreditCardNo |
| CreditCardCreateCIF |  |
| CreditCardEligibilityInquiry |  |
| CreditCardEligibilityProcess |  |
| CreditCardEligibilityRetrieveApplications |  |
| CreditCardEligibilityRetrieveTable |  |
| CreditCardEligibilityUpdate |  |
| CreditCardGetActivate | CreditCardNo |
| CreditCardGetAddress | CreditCardNo |
| CreditCardGetPayMethod | CreditCardNo |
| CreditCardHistory |  |
| CreditLimitIncrease | CreditCardNo |
| CreditCardLinkAccount | CreditCardNo |
| CreditCardNewPayMethod | CreditCardNo |
| CreditCardPay | CreditCardNo |
| CreditCardPendingTran | CreditCardNo |
| CreditCardPinReIssue | CreditCardNo |
| CreditCardReIssue | CreditCardNo - or all can be routed to cortex to re-issue under new product |
| CreditCardsList |  |
| CreditCardModify |  |
| CreditCardStatement | CreditCardNo |
| CreditCardStatementMerchant | MerchantID? |
| CreditCardStmtDetails | CreditCardNo |
| CreditCardStmtReq | CreditCardNo |
| CreditCardStmtReqDate | CreditCardNo |
| CreditCardSupplementary | CreditCardNo |
| CreditCardTransactions | CreditCardNo |
| CreditCardUnBlock | CreditCardNo |
| CreditCardApplicationInquiryShort | ApplicationNo |
| CreditCardApplicationCreate | All requests can be routed to cortex for new applications or source of the request or product requestd… |
| CreditCardApplicationApproveDecline | ApplicationNo |
| CreditCardApplicationUpdate | UniqueId/ApplicationNo |
| GetIndicativeFxRate |  |
| CreditCardPaymentAdvice |  |
| CreditCardSMSInquiry | CreditCardNo |
| CreditCardSMSUpdate |  |
| CreditCardSMSNotify | CreditCardNo |
| CreditCardLIRecordUpdate | CardNumber |
| CreditCardPinSelect | CreditCardNo |
| CreditCardIssue | CreditCardNo |
| CreditCarPreIssue | CreditCardNo |
| CreditCardPinChange | CardNumber |
| CreditCarPinChangeReversal | CardNumber |
| CreditCardReplacement | CreditCardNo |
| CreditCardSupplementaryCreate | CreditCardNo |
| CreditCardPrepaidApplicationCreate | UniqueId/Logo |
| CreditCardPrepaidApplicationUpdate | UniqueId/Logo |
| CreditCardPrepaidApplicationInquiryDetail | UniqueId/ApplicationNo |
| CreditCardChangeEmbossedName | CreditCardNo |
| CreditCardListByID | UniqueID/ID |
| UpdateParty | Id |
| EchoTest | N/A |
| PinDisplay |  |

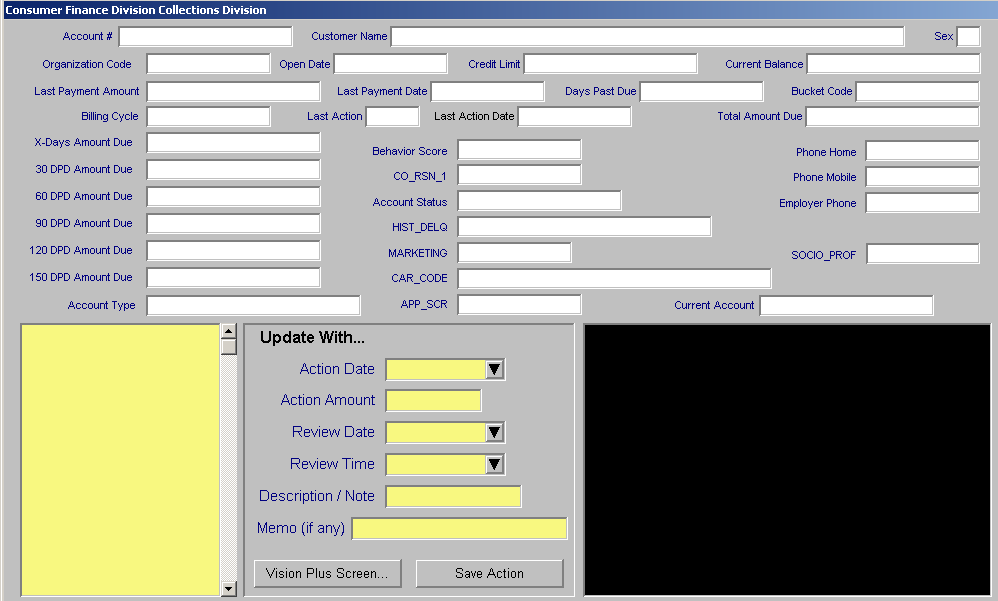
## 2.2 Predictive dialler new services

As part of the integration between the existing system and vision+ currently they are using screen scrapping to retrieve or to update some information in vision +.

In order to support the below functionalities without major changes to the flow we decide to provide two new services that will cover the required functionalities

-**Update Account Collection Status Service**

The update collection status will make a call to the collection system and update the account collection status in to the one chosen in the below screen.



Specifications for **Get Credit Card Account Details Service is as follows:**

<?xml version="1.0" encoding="UTF-8" standalone="no"?><wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:sch="http://com.emcrey.apxws/proto/apx" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://com.emcrey.apxws/proto/apx" targetNamespace="http://com.emcrey.apxws/proto/apx">

<wsdl:types>

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:pcm="http://com.emcrey.apxws/proto/apx" attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://com.emcrey.apxws/proto/apx">

<element name="updateCollectionStatusRequest">

<complexType>

<sequence>

<element name="MsgRqHdr">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="1" name="RqUID" type="string"/>

<element maxOccurs="1" minOccurs="1" name="SCId" type="string"/>

<element maxOccurs="1" minOccurs="1" name="FuncId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="RqMode" type="string"/>

<element maxOccurs="1" minOccurs="0" name="CustLangPref" type="string"/>

<element maxOccurs="1" minOccurs="0" name="TerminalId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="TerminalDesc" type="string"/>

<element maxOccurs="1" minOccurs="0" name="BankId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="BranchId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="BranchName" type="string"/>

<element maxOccurs="1" minOccurs="0" name="CustId">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="CustNum" type="string"/>

<element name="CardId">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="CardNum" type="string"/>

<element maxOccurs="1" minOccurs="0" name="CardType" type="string"/>

</sequence>

</complexType>

</element>

<element name="AcctId">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="AcctNum" type="string"/>

<element maxOccurs="1" minOccurs="0" name="AcctType" type="string"/>

</sequence>

</complexType>

</element>

<element name="POI">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="POINum" type="string"/>

<element maxOccurs="1" minOccurs="0" name="POIType" type="string"/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="0" name="UsrId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="NumOfRecs" type="string"/>

<element maxOccurs="1" minOccurs="0" name="Sec">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="Info" type="string"/>

<element maxOccurs="1" minOccurs="0" name="InfoType" type="string"/>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="0" name="Override">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="UsrId" type="string"/>

<element name="Sec">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="Info" type="string"/>

<element maxOccurs="1" minOccurs="0" name="InfoType" type="string"/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="0" name="AgentId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="ClientDt" type="string"/>

<element maxOccurs="1" minOccurs="0" name="Version" type="string"/>

<element maxOccurs="1" minOccurs="0" name="EchoData" type="string"/>

<element maxOccurs="1" minOccurs="0" name="OrgMsgId" type="string"/>

</sequence>

</complexType>

</element>

<element name="Body">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="1" name="DDAccountNumber" type="string"/>

<element maxOccurs="1" minOccurs="1" name="CCAccountNumber" type="string"/>

<element maxOccurs="1" minOccurs="1" name="COLFileNumber" type="string"/>

<element maxOccurs="1" minOccurs="1" name="VirtualPan" type="string"/>

<element maxOccurs="1" minOccurs="1" name="Clear" type="string"/>

<element maxOccurs="1" minOccurs="1" name="Encrypted" type="string"/>

<element maxOccurs="1" minOccurs="1" name="CustomerCode" type="string"/>

<element maxOccurs="1" minOccurs="1" name="NewStatusCode" type="string"/>

<element maxOccurs="1" minOccurs="1" name="ActionDate" type="string"/>

<element maxOccurs="1" minOccurs="1" name="ActionAmount" type="string"/>

<element maxOccurs="1" minOccurs="1" name="Reviewdate" type="string"/>

<element maxOccurs="1" minOccurs="1" name="ReviewTime" type="string"/>

<element maxOccurs="1" minOccurs="1" name="UserNote" type="string"/>

<element maxOccurs="1" minOccurs="0" name="UserMemo" type="string"/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element name="updateCollectionStatusResponse">

<complexType>

<sequence>

<element name="MsgRsHdr">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="1" name="StatusCode" type="string"/>

<element maxOccurs="1" minOccurs="1" name="StatusDesc" type="string"/>

<element name="HostStatus">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="HostName" type="string"/>

<element maxOccurs="1" minOccurs="0" name="HostStatusCode" type="string"/>

<element maxOccurs="1" minOccurs="0" name="HostStatusDesc" type="string"/>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="1" name="RqUID" type="string"/>

<element maxOccurs="1" minOccurs="0" name="SPRefNum" type="string"/>

<element maxOccurs="1" minOccurs="0" name="MsgRecDt" type="string"/>

<element maxOccurs="1" minOccurs="0" name="EchoData" type="string"/>

<element maxOccurs="1" minOccurs="0" name="BranchName" type="string"/>

<element maxOccurs="1" minOccurs="0" name="AgentId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="ClientDt" type="string"/>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="0" name="Body">

<complexType>

<sequence>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

</schema>

</wsdl:types>

<wsdl:message name="updateCollectionStatusResponse">

<wsdl:part element="tns:updateCollectionStatusResponse" name="updateCollectionStatusResponse">

</wsdl:part>

</wsdl:message>

<wsdl:message name="updateCollectionStatusRequest">

<wsdl:part element="tns:updateCollectionStatusRequest" name="updateCollectionStatusRequest">

</wsdl:part>

</wsdl:message>

<wsdl:portType name="UpdateCollectionStatus">

<wsdl:operation name="updateCollectionStatus">

<wsdl:input message="tns:updateCollectionStatusRequest" name="updateCollectionStatusRequest">

</wsdl:input>

<wsdl:output message="tns:updateCollectionStatusResponse" name="updateCollectionStatusResponse">

</wsdl:output>

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name="UpdateCollectionStatusSoap11" type="tns:UpdateCollectionStatus">

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<wsdl:operation name="updateCollectionStatus">

<soap:operation soapAction=""/>

<wsdl:input name="updateCollectionStatusRequest">

<soap:body use="literal"/>

</wsdl:input>

<wsdl:output name="updateCollectionStatusResponse">

<soap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="UpdateCollectionStatusService">

<wsdl:port binding="tns:UpdateCollectionStatusSoap11" name="UpdateCollectionStatusSoap11">

<soap:address location="http://localhost:8080/apex-ws-core/updateCollectionStatus/"/>

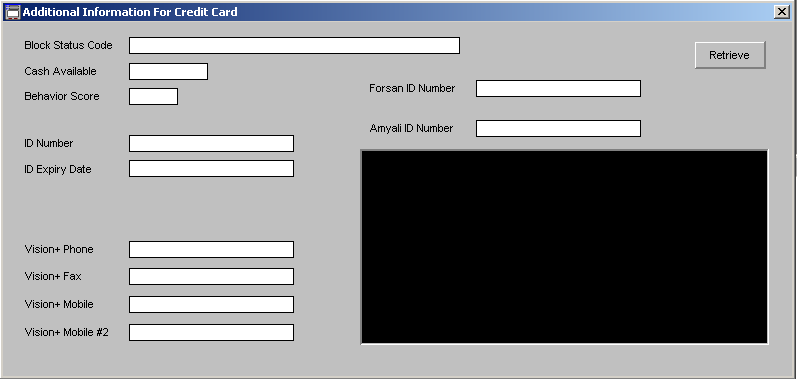
</wsdl:port>

</wsdl:service>

</wsdl:definitions>

**Get Credit Card Account Details Service**

The get credit card account details is a web-services that will be used to get the information from the credit card system in order to populate the below fields.



Specification for the **Get Credit Card Account Details Service**

<?xml version="1.0" encoding="UTF-8" standalone="no"?><wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:sch="http://com.emcrey.apxws/proto/apx" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://com.emcrey.apxws/proto/apx" targetNamespace="http://com.emcrey.apxws/proto/apx">

<wsdl:types>

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:pcm="http://com.emcrey.apxws/proto/apx" attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://com.emcrey.apxws/proto/apx">

<element name="creditCardAccountInquiryRequest">

<complexType>

<sequence>

<element name="MsgRqHdr">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="1" name="RqUID" type="string"/>

<element maxOccurs="1" minOccurs="1" name="SCId" type="string"/>

<element maxOccurs="1" minOccurs="1" name="FuncId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="RqMode" type="string"/>

<element maxOccurs="1" minOccurs="0" name="CustLangPref" type="string"/>

<element maxOccurs="1" minOccurs="0" name="TerminalId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="TerminalDesc" type="string"/>

<element maxOccurs="1" minOccurs="0" name="BankId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="BranchId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="BranchName" type="string"/>

<element maxOccurs="1" minOccurs="0" name="CustId">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="CustNum" type="string"/>

<element name="CardId">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="CardNum" type="string"/>

<element maxOccurs="1" minOccurs="0" name="CardType" type="string"/>

</sequence>

</complexType>

</element>

<element name="AcctId">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="AcctNum" type="string"/>

<element maxOccurs="1" minOccurs="0" name="AcctType" type="string"/>

</sequence>

</complexType>

</element>

<element name="POI">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="POINum" type="string"/>

<element maxOccurs="1" minOccurs="0" name="POIType" type="string"/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="0" name="UsrId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="NumOfRecs" type="string"/>

<element maxOccurs="1" minOccurs="0" name="Sec">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="Info" type="string"/>

<element maxOccurs="1" minOccurs="0" name="InfoType" type="string"/>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="0" name="Override">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="UsrId" type="string"/>

<element name="Sec">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="Info" type="string"/>

<element maxOccurs="1" minOccurs="0" name="InfoType" type="string"/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="0" name="AgentId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="ClientDt" type="string"/>

<element maxOccurs="1" minOccurs="0" name="Version" type="string"/>

<element maxOccurs="1" minOccurs="0" name="EchoData" type="string"/>

<element maxOccurs="1" minOccurs="0" name="OrgMsgId" type="string"/>

</sequence>

</complexType>

</element>

<element name="Body">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="1" name="DDAccountNumber" type="string"/>

<element maxOccurs="1" minOccurs="1" name="CCAccountNumber" type="string"/>

<element maxOccurs="1" minOccurs="1" name="COLFileNumber" type="string"/>

<element maxOccurs="1" minOccurs="0" name="VirtualPan" type="string"/>

<element maxOccurs="1" minOccurs="0" name="Clear" type="string"/>

<element maxOccurs="1" minOccurs="0" name="Encrypted" type="string"/>

<element maxOccurs="1" minOccurs="0" name="CustomerCode" type="string"/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

<element name="creditCardAccountInquiryResponse">

<complexType>

<sequence>

<element name="MsgRsHdr">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="1" name="StatusCode" type="string"/>

<element maxOccurs="1" minOccurs="1" name="StatusDesc" type="string"/>

<element name="HostStatus">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="HostName" type="string"/>

<element maxOccurs="1" minOccurs="0" name="HostStatusCode" type="string"/>

<element maxOccurs="1" minOccurs="0" name="HostStatusDesc" type="string"/>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="1" name="RqUID" type="string"/>

<element maxOccurs="1" minOccurs="0" name="SPRefNum" type="string"/>

<element maxOccurs="1" minOccurs="0" name="MsgRecDt" type="string"/>

<element maxOccurs="1" minOccurs="0" name="EchoData" type="string"/>

<element maxOccurs="1" minOccurs="0" name="BranchName" type="string"/>

<element maxOccurs="1" minOccurs="0" name="AgentId" type="string"/>

<element maxOccurs="1" minOccurs="0" name="ClientDt" type="string"/>

</sequence>

</complexType>

</element>

<element maxOccurs="1" minOccurs="0" name="Body">

<complexType>

<sequence>

<element maxOccurs="1" minOccurs="0" name="AccnoStatusCode" type="string"/>

<element maxOccurs="1" minOccurs="0" name="CashAvailable" type="string"/>

<element maxOccurs="1" minOccurs="0" name="BehaviorScore" type="string"/>

<element maxOccurs="1" minOccurs="0" name="IdNumber" type="string"/>

<element maxOccurs="1" minOccurs="0" name="idExpiryDate" type="string"/>

<element maxOccurs="1" minOccurs="0" name="ForsanIdNumber" type="string"/>

<element maxOccurs="1" minOccurs="0" name="AmyaliIdNumber" type="string"/>

<element maxOccurs="1" minOccurs="0" name="PhoneNumber" type="string"/>

<element maxOccurs="1" minOccurs="0" name="FaxNumber" type="string"/>

<element maxOccurs="1" minOccurs="0" name="MobileNumber" type="string"/>

<element maxOccurs="1" minOccurs="0" name="MobileNumber2" type="string"/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

</element>

</schema>

</wsdl:types>

<wsdl:message name="creditCardAccountInquiryResponse">

<wsdl:part element="tns:creditCardAccountInquiryResponse" name="creditCardAccountInquiryResponse">

</wsdl:part>

</wsdl:message>

<wsdl:message name="creditCardAccountInquiryRequest">

<wsdl:part element="tns:creditCardAccountInquiryRequest" name="creditCardAccountInquiryRequest">

</wsdl:part>

</wsdl:message>

<wsdl:portType name="CreditCardAccountInquiry">

<wsdl:operation name="creditCardAccountInquiry">

<wsdl:input message="tns:creditCardAccountInquiryRequest" name="creditCardAccountInquiryRequest">

</wsdl:input>

<wsdl:output message="tns:creditCardAccountInquiryResponse" name="creditCardAccountInquiryResponse">

</wsdl:output>

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name="CreditCardAccountInquirySoap11" type="tns:CreditCardAccountInquiry">

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<wsdl:operation name="creditCardAccountInquiry">

<soap:operation soapAction=""/>

<wsdl:input name="creditCardAccountInquiryRequest">

<soap:body use="literal"/>

</wsdl:input>

<wsdl:output name="creditCardAccountInquiryResponse">

<soap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="CreditCardAccountInquiryService">

<wsdl:port binding="tns:CreditCardAccountInquirySoap11" name="CreditCardAccountInquirySoap11">

<soap:address location="http://localhost:8080/apex-ws-core/creditCardAccountInquiry/"/>

</wsdl:port>

</wsdl:service>

</wsdl:definitions>

# Batch Files.

Batch files will need to be split and or merged depending on whether it is inbound or outbound from the various applications. eMCrey will develop batch programs to split and merge files. All AFS output files will come to Cortex before being applied to other systems (to be merged) and all output files from NCB will come through Cortex (to be split)

## 3.1 Complete list of Batch Files

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Extract Name** | **VISION+ Cortex** | | Action to Be Taken | |
|  |  | **Source system** | **Destination system** | Merge/Split/Two files | |
| K024 | Montly AFSDMERCHNT (CC Merchants) | Vision+ / Cortex | DWH | Merge |  |
| K009 | Daily AFSDACQTXN (Non-NCB Cards transactions) | Vision+ / Cortex | DWH | Merge |  |
| K010 | Daily AFSDCLIENT (CC Customers) | Vision+ / Cortex | DWH | Merge |  |
| K011 | Daily AFSDCLIENTX (CC Customers) | Vision+ / Cortex | DWH | Merge |  |
| K013 | Daily AFSDPO10 (CC Card Production) | Vision+ / Cortex | DWH | Merge |  |
| K012 | Daily AFSDMERCHNT (CC Merchants) | Vision+ / Cortex | DWH | Merge |  |
| K014 | Daily AFSDACCT (CC Accounts) | Vision+ / Cortex | DWH | Merge |  |
| K015 | Daily AFSDISSTXN (CC Transactions) | Vision+ / Cortex | DWH | Merge |  |
| K016 | Daily AFSCDMF1 (CC Applications) | Vision+ / Cortex | DWH | Merge |  |
| K017 | Daily AFSDAUTH (CC Authorization) | Vision+ / Cortex | DWH | Merge |  |
| K018 | Daily AFSDCARDS (CC Cards) | Vision+ / Cortex | DWH | Merge |  |
| K019 | Daily AFSDCTAOCVI (CC Collections) | Vision+ / Cortex | DWH | Merge |  |
| K020 | Daily T\_CREDIT\_CARD\_RESCHEDULE (CC rescheduling) | Vision+ / Cortex | DWH | Merge |  |
| K021 | Montly AFSDACQTXN (Non-NCB Cards Transactions) | Vision+ / Cortex | DWH | Merge |  |
| K022 | Montly AFSDCLIENT (CC customers) | Vision+ / Cortex | DWH | Merge |  |
| K023 | Monthly AFSDCLIENTX (CC Customers) | Vision+ / Cortex | DWH | Merge |  |
| K025 | Montlty AFSDPO10 (CC Card Production) | Vision+ / Cortex | DWH | Merge |  |
| K026 | Montly AFSDACCT (CC Accounts) | Vision+ / Cortex | DWH | Merge |  |
| K027 | Monthly AFSDISSTXN (CC Transactions) | Vision+ / Cortex | DWH | Merge |  |
| K028 | Montlhly AFSCDMF1 (CC Applications) | Vision+ / Cortex | DWH | Merge |  |
| K029 | Montly AFSDAUTH (CC Authorizations) | Vision+ / Cortex | DWH | Merge |  |
| K030 | Monthly AFSDCARDS (CC Cards) | Vision+ / Cortex | DWH | Merge |  |
| K031 | Monthtly AFSDAMIH\_M (CC Summary Amih) | Vision+ / Cortex | DWH | Merge |  |
| K032 | Monthly AFSDAMIH\_V (CC Summary Amih) | Vision+ / Cortex | DWH | Merge |  |
| K033 | Monthtly AFSCCCLI (CC Limit) | Vision+ / Cortex | DWH | Merge |  |
| K034 | Montly AFSDCTAOCVI (Collections) | Vision+ / Cortex | DWH | Merge |  |
| K035 | Daily xlscamp (XLS) | Vision+ / Cortex | DWH | Merge |  |
| K036 | Daily xlsmerch (XLS merchant) | Vision+ / Cortex | DWH | Merge |  |
| K037 | Daily xlspoints (XLS Points) | Vision+ / Cortex | DWH | Merge |  |
| K038 | Daily xlstxnbal (XLS Balance) | Vision+ / Cortex | DWH | Merge |  |
| K039 | Daily xlstxndet (XLS Daily transactions details) | Vision+ / Cortex | DWH | Merge |  |
| K040 | Daily xlstxnhdr (XLS Header) | Vision+ / Cortex | DWH | Merge |  |
| K001 | SADAD - Bill upload to SADAD |  | Vision+ / Cortex | Split |  |
| K002 | SADAD - Payment Download |  | Vision+ / Cortex | Split |  |
| K003 | One View |  | Vision+ / Cortex | Split |  |
| K004 | ccbm0200 - credit card payment | Bancs | Vision+ / Cortex | Split |  |
| K005 | ccbd0310 - credit card delinquent details | Bancs | Vision+ / Cortex | Split |  |
| K006 | MPAC-BCOM-IN | Bancs | Vision+ / Cortex | Split |  |
| K007 | MPAC-PORTE-IN | Bancs | Vision+ / Cortex | Split |  |
| K008 | Dealer Payment | Bancs | Vision+ / Cortex | Split |  |
| K041 | OBC extract | Base24 | Vision + Cortex | Split |  |
| K042 | Socio Codes File Update |  | Vision+ / Cortex | Split |  |
| K043 | Segment File Update |  | Vision+ / Cortex | Split |  |
| K044 | Online Payment File |  | Vision+ / Cortex | Split |  |
| K045 | Branch Payment File |  | Vision+ / Cortex | Split |  |
| K046 | Mass Posting File |  | Vision+ / Cortex | Split |  |
| K047 | TeleSales File |  | Vision+ / Cortex | Split |  |
| K048 | Batch Application Import |  | Vision+ / Cortex | Split |  |
| K049 | Bulk Issuance Files |  | Vision+ / Cortex | Split |  |
| K050 | KYC and ID Expiry |  | Vision+ / Cortex | Split |  |
| K051 | Direct Debit Result |  | Vision+ / Cortex | Split |  |
| K052 | Fee Quotes Update |  | Vision+ / Cortex | Split |  |
| K053 | SMS File |  | Vision+ / Cortex | Split |  |
| K054 | Reports Exracts |  | Vision+ / Cortex | Split |  |
|  |  |  |  |  |  |

## 3.2 Batch Files handling during Co-existence of VISION+ and Cortex

## 3.2.1 Out Batch File Processing

For the files generated from the bank systems and sent to AFS for processing need to be updated in order not to send to AFS information which not hosted on their system which are related to the new bins or to the migrated one in the future

Below is the option that we will follow without impacting the AFS process

1. Option The system generating the credit card files will continue to generate one file including all the credit card information and all will be sent to CORTEX where a split process is developed and is responsible of scanning the file and generating the below two files

* First File include only data for cards hosted in AFS
* Second File include the data for cards hosted in Cortex

This option does not require any changes in the system generating the files and in CORTEX a new process will be added to do the split job

## 3.2.2 IN Batch File Processing

For the files generated from VISION+ and loaded to the bank systems the below option will be implemented.

1. Option

Each system CORTEX and VISION+ will generate separate file a merge process will be developed to merge the 2 files in one file

The option II does not require any changes in the system processing the files and in CORTEX a new process will be added to do the merge job

Please check the all files list specific to each channels – MW, Base24, DWH, ReconSS, BANCS, IVR, XLS, PMA



## 3.3 Smart Hub

Two MQ Queues will be defined one inbound and one outbound to update Customer Information only. This is to ensure that any customer information updated directly on Cortex system will be synchronized with other systems that hold Customer information within the Bank.

The Queue name from SmartHub to Cortex will be defined as MQCortexout

The Queue name from Cortex to Smarthub will be defined as MQSmartHUBin.

At this stage it is envisaged that only address changes and telephone number changes will be sent to and received from SmartHub. No other customer information changes will be applied.

## 3.4 Identify systems that will use the new BINS

Identify in which system the new bins and products should be defined for smooth rollout .

* For cortex credit card new BINs– Logo 491 & Logo- 531 would be considered.
* Add the BIN in Affina for central and instant both. configured - Yes.
* Load BIN in Base24- configured - Yes.
* Load logo parameters table in MW for below new BINs – For instant issuance.
* Update the BIN information in MW E- portal
* Update the BIN Information in required parameters in DWH
* Upload new images for below in Beam10 & UEB channel.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BIN | ORG | Logo | Billing cycles | PCT | Name |
| 521031 | 007 | 568 | 12 | 001, 150,300,400 | Master Platinum |
| 419991 | 007 | 460 | 12 | 001, 150,300,400 | Visa Platinum |

Other paramters can be use as like 523968 & 412113.

## 

## 3.5 Control M Changes

The Control M will have some changes related to file transfers and for job scheduling which need to be identified

## 

## 3.6 Cordys

## 3.6 Middleware