Internet Banking

Technical Specifications Documentation

Table of Contents

[General Information 4](#_Toc277607488)

[Introduction 4](#_Toc277607489)

[Purpose 4](#_Toc277607490)

[Scope 4](#_Toc277607491)

[System Overview 5](#_Toc277607492)

[System Component Overview 5](#_Toc277607493)

[Application Design Overview 6](#_Toc277607494)

[General Specification 8](#_Toc277607495)

[Framework 8](#_Toc277607496)

[Apache Struts 8](#_Toc277607497)

[Hibernate 9](#_Toc277607498)

[Authentication (UPass) 11](#_Toc277607499)

[UPass Security Specification 12](#_Toc277607500)

[Configuration 13](#_Toc277607501)

[Integration (JMS) 14](#_Toc277607502)

[What is JMS 14](#_Toc277607503)

[JMS Platform 14](#_Toc277607504)

[MPI JMS Mechanism 14](#_Toc277607505)

[WebLogic JMS Architecture 14](#_Toc277607506)

[MPI JMS Architect 15](#_Toc277607507)

[MPI JMS Message Design 16](#_Toc277607508)

[JMS Message Process Explanation 16](#_Toc277607509)

[Module’s Technical Specifications 18](#_Toc277607510)

[Introduction (Module listing and breakdown) 18](#_Toc277607511)

[IBSS/MSS 18](#_Toc277607512)

[Retail Internet Banking 18](#_Toc277607513)

[Business Internet Banking 20](#_Toc277607514)

[Common Internet Banking Feature 21](#_Toc277607515)

[Portal Access 22](#_Toc277607516)

[First Time Login 22](#_Toc277607517)

[Reset Password 25](#_Toc277607518)

[Login 27](#_Toc277607519)

[Logout 29](#_Toc277607520)

[Error Logout 31](#_Toc277607521)

[Account Enquiry 32](#_Toc277607522)

[Account Summary 32](#_Toc277607523)

[Account Details 34](#_Toc277607524)

[Internet Transaction History 36](#_Toc277607525)

[Transaction History 38](#_Toc277607526)

[Cheque Management 40](#_Toc277607527)

[Reorder Cheque Book 40](#_Toc277607528)

[Cheque Warehousing 42](#_Toc277607529)

[Cheque Status Enquiry / Stop Cheque 44](#_Toc277607530)

[Cheque Cutting Enquiry 47](#_Toc277607531)

[Account Maintenance 49](#_Toc277607532)

[Link Account 49](#_Toc277607533)

[Unlink Account 51](#_Toc277607534)

[Fund Transfer 53](#_Toc277607535)

[Own Account Transfer 53](#_Toc277607536)

[3rd Party Account Transfer 56](#_Toc277607537)

[Interbank Account Transfer 59](#_Toc277607538)

[Favorite Maintenance 60](#_Toc277607539)

[Future Transfer 61](#_Toc277607540)

[Cash Card Top Up 62](#_Toc277607541)

# General Information

## Introduction

Internet banking system can be segregated to three different applications, Internet banking Administration Application, Retail Internet Banking Application and Business Internet Banking Application. Which Administration Application governs or controls how Retail Internet Banking and Business Internet Banking applications work. Administration application…… Retail Internet Banking…… Business Internet Banking……

## Purpose

The main purpose of this document is to cover system design and technical specification of the development for internet banking system. It will describe the technical design……

## Scope

This document will cover the below mention:

* System architecture design overview
* System development frameworks
* Database specification
* Integration specification
* System authentication specification
* Common application control functions
* Application modules’ specifications

## System Overview

MPI Internet Banking system consists of three major application; Internet Banking Support System (IBSS/MSS), Retail Internet Banking (RIB) and Business Internet Banking (BIB); and sub applications such like UPass Authentication System and JMS for Core Banking integration.

The three major applications are incorporated to each other via Database. All these applications are sharing the same Database and Schema. RIB and BIB applications are communicating with JMS via JMS Web Service Client created mainly for JMS integration, whilst JMS will handle all request and response messages and messages queuing for Core Banking Host integration.



Figure System Overview

### System Component Overview

The MPI Internet Banking system is staged with a 2 layers of service components, the Application Control Layer and Integration Control Layer.

The Application Control Layer is to control the system page flows and the integration for the page flow to different service components. This layer is using the Model-View-Controller (MVC) software architecture pattern. The service components consist of different type of services to support the system, like the control of page navigation, authentication (UPASS), user access matrix, etc., through individual wrapper components, like Authentication Component, Data Access Component, Business Implementation Component, Host Integration Component and External Integration Component.

The Integration Control layer is build using Java Message Service (JMS); see JMS Section for more details on how JMS shall integrate with the MPI Host System.

Figure Application Components Overview

### Application Design Overview

The below diagram is the overview of the MPI Internet Banking application architecture framework. The system design is to componentized the system into different service components and also using the MVC architect pattern to isolate the “domain logic” (business/application logic) from the user interface (presentation layer) permitting independent development, testing and maintenance of each.

For the MPI project, the Struts 1.2 framework is adapted as the MVC model implementation. To complete the system design, the Business Service Wrapper layer is also created, consists of business logics, to act as the communication layer between the Struts Action layer and other backend services.

Figure 3 Application Design Overview

# General Specification

## Framework

MPI Internet Banking system is build using Apache Struts framework, an open-source web application framework for developing Java EE web applications. It uses and extends the Java Servlet API to encourage developers to adopt a model-view-controller (MVC) architecture. For application data access, MPI Internet Banking applications are built on top of Hibernate, an [object-relational mapping](http://en.wikipedia.org/wiki/Object-relational_mapping) (ORM) library for the [Java](http://en.wikipedia.org/wiki/Java_(programming_language)) language, providing a [framework](http://en.wikipedia.org/wiki/Software_framework) for mapping an [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming) [domain model](http://en.wikipedia.org/wiki/Domain_model) to a traditional [relational database](http://en.wikipedia.org/wiki/Relational_database).

### Apache Struts

Apache Struts is an open source framework used for developing J2EE web applications using Model View Controller (MVC) design pattern. It uses and extends the Java Servlet API to encourage developers to adopt an MVC architecture. Struts framework provides three key components:

* A **request** handler provided by the application developer that is used to map to a particular URI.
* A **response** handler which is used to transfer the control to another resource which will be responsible for completing the response.
* A **tag library** which helps developers to create the interactive form based applications with Java Server Pages.

The below diagram depicts the architecture of MVC system and the design to decouple different layers at the Business functions layer.

All Action classes will have methods to call from DAO zone. These DAO will have the real code to connect to the external systems like Database, CICS, RSA, etc. Optionally, all connection objects shall be received from centralized Class; so that any new connection objects or modifications shall be controlled from one class. All data returned from DAO are in normal Java Bean Objects, not any system specific object. This will help in easy integration with any system in future stages.



Figure MPI Struts Application Overview

### Hibernate

Hibernate is an object-relational mapping (ORM) library for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions.

Mapping Java classes to database tables is accomplished through the configuration of an XML file or by using Java Annotations. When using an XML file, Hibernate can generate skeletal source code for the persistence classes. This is unnecessary when annotation is used. Hibernate can use the XML file or the annotation to maintain the database schema.

Facilities to arrange one-to-many and many-to-many relationships between classes are provided. In addition to managing association between objects, Hibernate can also manage reflexive associations where an object has a one-to-many relationship with other instances of its own type.

Hibernate supports the mapping of custom value types. This makes the following scenarios possible:

* Overriding the default SQL type that Hibernate chooses when mapping a column to a property.
* Mapping Java Enum to columns as if they were regular properties.
* Mapping a single property to multiple columns.

## Authentication (UPass)

MPI Internet Banking System is using UPass Authentication API for to manage user access credential. UPass is a standalone API builds on Java Platform, the below diagram illustrates UPass authentication process flow:

Figure UPass Application Process Flow

UPass Authentication Features as Below:

* Encryption methods
  + SHA-1
  + 3DES
* Default Super Admin username and password
  + Configured during installation
* Password Maintenance
  + Password expire duration
  + Non-reuse for number of generation
  + Password composition
  + Features
  + addUser
  + deleteUser
  + changePassword
  + verifyPassword

### UPass Security Specification

|  |  |  |
| --- | --- | --- |
| User ID and Password Control | Minimum Requirements | Remark |
| 1. User ID | Single or combination of numeric, alpha or alphanumeric. | The rule is not parameterized. |
| 1. Maximum password expiration | **90 days** (The system will force a reset after this period has elapsed) | System wide support and can be parameterized via the system configuration file |
| 1. Minimum password length | **8** characters | This is standard feature. |
| 1. Maximum failed log-in attempts | **3 times.** (The User ID will subsequently be locked (disabled)) | The maximum number of attempts can be parameterized via the system configuration file. If the User ID is locked, the user must send a request to the administrator to reset the password. The application will force the user to change their password upon login. |
| 1. Inactive/Dormant ID for more than 90 days | User ID is automatically disabled by the system | The number of days can be parameterized via the system configuration file. |
| 1. Duplicate password control | For a minimum of **6** generations, the same passwords cannot be reused. | The number of generations can be parameterized via the system configuration file. |
| 1. Last sign on information displayed | The last signed on details will be displayed to the user upon logging into the system | This is standard feature. |
| 1. First time sign on must force a password change | First time sign on user must be forced to change the defaulted password. | This is standard feature. |
| 1. Changing of passwords by user | Users may change their passwords anytime they wish | This is standard feature. |
| 1. Password composition | Must be combination of alphanumeric, special characters, lower/upper case (if possible) | This is standard feature.  The password must be combination of at least 1 alphabet and 1 numeric value and is case sensitive. |
| 1. Deletion of User IDs from application | User IDs that are obsolete must be able to be deleted (instead of just suspended) from the application | The same user id cannot be reused even if deleted. The deleted user will not show on user enquiry screen, but the record will still be stored in the database. |
| 1. Password storage | Requires Encryption | Combination of Login ID and Password is encrypted using SHA-1 before being stored into Database. The hashed value will be used for authentication |

### Configuration

|  |  |  |
| --- | --- | --- |
| Configuration | Parameter | Remark |
| Maximum allowed password input error | PASSWORD\_MAX\_ERROR | Default is 3 allowable password input error |
| Password history storage | PASSWORD\_GENERAION | Default is 6 generations (Database maximum capacity) |
| Password expiry day | PASSWORD\_EXPIRY\_DAY | Default is 90 days password expiry before next reset |
| Minimum password length | PASSWORD\_MIN\_LENGTH | Default is 8 which is similar to application validation |
| Password complexity | PASSWORD\_COMPLEXITY | Default is 0. Application validates alphanumeric. Complexity of 1 requires a combination of lower and upper characters, numbers and special characters |
| Maximum days to be classified as dormant | MAX\_DORMANT\_DAYS | Default is 90 days |

## Integration (JMS)

### What is JMS

The Java Message Service (JMS) API is a Java Message Oriented Middleware API (MOM) for sending messages between two or more clients. JMS is a part of the Java Platform, Enterprise Edition, and is defined by a specification developed under the Java Community Process as JSR 914. It is a messaging standard that allows application components based on the Java 2 Platform, Enterprise Edition (J2EE) to create, send, receive, and read messages. It allows the communication between different components of a distributed application to be loosely coupled, reliable, and asynchronous.

### JMS Platform

Implementing JMS is platform specific configuration. For this MPI project, the JMS design is based on Oracle WebLogic 10.

### MPI JMS Mechanism

Figure 4 shows the overview design of the JMS component. The JMS component consists of 2 subcomponents, JMS Client (also called Producer) and JMS Server (Consumer / Producer).

JMS Client will receive a request, drop the request message to the queue, wait for the response message and return the response message once obtained before timeout. The queue request is demonstrated in Figure 4 start from step 1.1 and completed of receiving response at step 1.16.

JMS Server will read a message from queue, process it, get the response info, construct the response message and drop the response message to the queue. The process message mechanism is demonstrated in Figure 4 start from step 2.1 and completed of dropping response message at step 2.4.

### WebLogic JMS Architecture

Major components of the WebLogic JMS Server architecture:-

* JMS servers that can host a defined set of modules and any associated persistent storage that resides on a WebLogic Server instance.
* JMS modules contain configuration resources (such as queues, topics, and connections factories) and are defined by XML documents that conform to the <http://xmlns.oracle.com/weblogic/weblogic-jms/1.0/weblogic-jms.xsd> schema.
* Client JMS applications that either produce messages to destinations or consume messages from destinations.
* JNDI (Java Naming and Directory Interface), which provides a resource lookup facility. JMS resources such as connection factories and destinations are configured with a JNDI name. The runtime implementations of these resources are then bound into JNDI using the given names.
* WebLogic persistent storage (file store or JDBC-accessible) for storing persistent message data.



Figure WebLogic JMS Architecture Diagram

### MPI JMS Architect

There are 3 JMS Client components and 2 queues are designed to handle MPI Internet Banking messages.

The 2 queues are the Request Queue and the Response Queue. The Request Queue is for request messages whereas the Response Queue is designed for response messages.

Each of the JMS Client has its own task.

JMS Client 1: is designed to receive a request message from caller, through Web Service, and drop the message to the Request Queue.

JMS Client 2: is designed to pickup response messages from the Response Queue, also through a Web Service call from the caller.

JMS Client 3: is designed to pickup messages from the Request Queue, process the request, construct the response message and drop the response message to the Response Queue.



Figure MPI JMS Architecture Diagram

### MPI JMS Message Design

Each JMS message is identified by a **unique message id**. A unique id shall be generated and assigned to the Enterprise Java Bean (EJB) at the JMS Client stage. This id will be used for the entire lifecycle of the queue process. At the JMS Server stage, once a message is being processes, the same message id shall be used to construct the response message before drop it to the queue.

### JMS Message Process Explanation

The diagram below shows that the service layer will drop a request message to the JMS Response Queue, through the Request Web Service call, and then wait at the JMS Response Queue for the response, through the Response Web Service call. Once the response message is received, the message will be processed at the M2U Service Layer.

At the JMS Server layer (Process layer), the Request MDB will pick up a message from the JMS Request Queue, convert it to host message format and drop the message to CICS through socket. The Request MCB will get the response from CICS and then drop the message to the JMS Response Queue to complete the task.

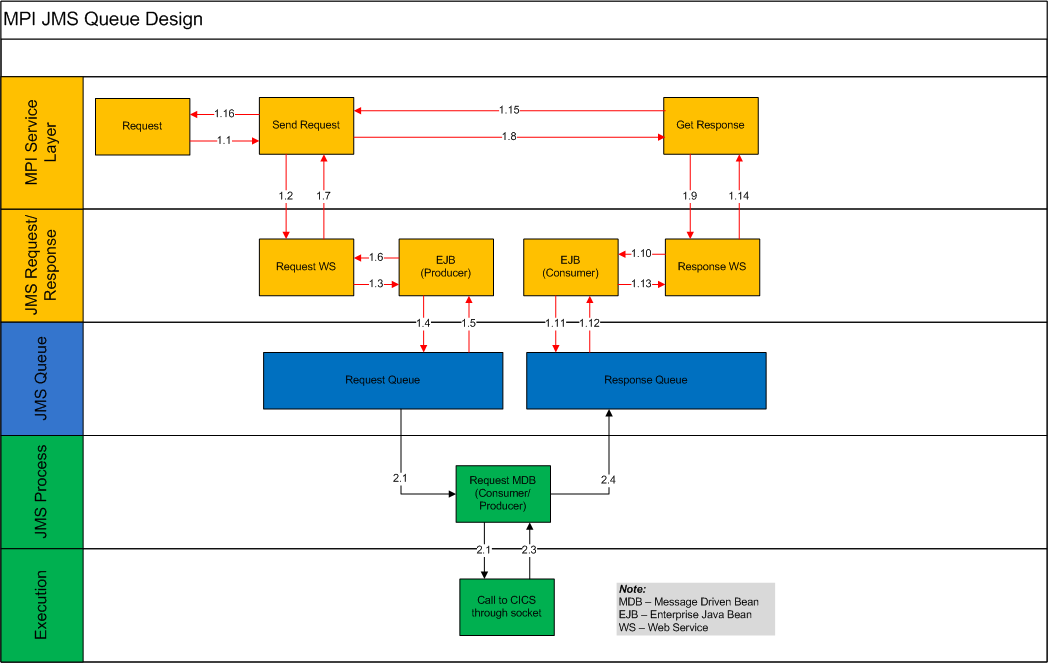


Figure JMS Queue Process Diagram

# Module’s Technical Specifications

## Introduction (Module listing and breakdown)

This section explains the technical specifications of each module and its sub modules which are included in MPI Internet Banking System deliverables.

### IBSS/MSS

* **MSS Portal Access**
  + First Time Login/Reset Password Login (After Reset Password)
  + Access Login
  + Logout
  + Error Logout
* **MSS Administration**
  + MSS User Management
  + MSS User Group Management
  + Content Management
  + Audit Log
  + Profile Maintenance
* **RIB Administration**
  + Navigation
  + RIB User Management
  + RIB User Group Management
  + RIB Transaction Enquiry
* **BIB Administration**
  + Navigation
  + BIB Company Management
    - Company User Management
  + BIB User Group Management
  + BIB Transaction Enquiry

### Retail Internet Banking

* **Portal Access**
  + Registration
  + Reset Password
  + Login
  + Logout
  + Error Logout
* **Account Enquiry**
  + CASA Account Summary
    - Account Details
    - Internet Transaction History
    - Transaction History
    - Cheque Management
      * Reorder Cheque Book
      * Cheque Warehousing
      * Cheque Status Enquiry
      * Stop Cheque
  + Loan Account Summary
* **Account Maintenance**
  + Link Account
  + Unlink Account
* **Fund Transfer**
  + Own Account Transfer
  + 3rd Party Account Transfer
    - Open Transfer
    - Favorite Transfer
  + Interbank Account Transfer
    - Open Transfer
    - Favorite Transfer
  + Favorite Maintenance
    - Add
    - Edit
    - Delete
  + Future Transfer
    - Future Transfer History
    - Cancel Future Transfer
  + Cash Card Top Up
    - Top Up
* **Bill Payment**
  + Open Payment
  + Favorite Payment
  + Favorite Maintenance
    - Add
    - Edit
    - Delete
  + Future Payment
    - Future Payment History
    - Cancel Future Payment
* **Time Deposit**
  + TD Account Enquiry
  + TD Placement

### Business Internet Banking

* **Portal Access**
  + Registration
  + Reset Password
  + Login
  + Logout
  + Error Logout
* **Account Enquiry**
  + CASA Account Summary
    - Account Details
    - Internet Transaction History
    - Transaction History
    - Cheque Management
      * Reorder Cheque Book
      * Cheque Warehousing
      * Cheque Status Enquiry
      * Stop Cheque
      * Cheque Cutting
  + Loan Account Summary
* **Account Maintenance**
  + Link Account
  + Unlink Account
* **Fund Transfer**
  + Own Account Transfer
  + 3rd Party Account Transfer
    - Open Transfer
    - Favorite Transfer
  + Interbank Account Transfer
    - Open Transfer
    - Favorite Transfer
  + Favorite Maintenance
    - Add
    - Edit
    - Delete
  + Future Transfer
    - Future Transfer History
    - Cancel Future Transfer
  + Cash Card Top Up
    - Top Up
* **Bill Payment**
  + Open Payment
  + Favorite Payment
  + Favorite Maintenance
    - Add
    - Edit
    - Delete
  + Future Payment
    - Future Payment History
    - Cancel Future Payment
* **Time Deposit**
  + TD Account Enquiry
  + TD Placement

### Common Internet Banking Feature

* **Navigation**
* **QuickLink**
* **Single Signon**
* **Session Handling**
* **System Logging**
* **Error Handling**
* **Service Info**

## Portal Access

### First Time Login

**Module Overview**

First time login function allow customer to register his/her Internet Banking account by creating desire Username and Password.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbFTL** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Terms and Condition | | | | | |
| agreeTC | T&C agreement by user | String | Button | - | - |
| Registration Verification | | | | | |
| accessNo | IB access no/CIF no | String | Text | Require  Alphanumeric | Y |
| fTac | FTAC | String | Password | Require  Numeric | N |
| Registration Details | | | | | |
| username | User desire username | String | Text | Require  Alphanumeric  Min Length = 6  Max Length = 16 | Y |
| password1 | User desire password | String | Password | Require  Alphanumeric  Min Length = 8  Max Length = 12 | Y |
| password2 | User reenter password | String | Password | Require  Alphanumeric  Min Length = 8  Max Length = 12  Match password1 | N |
| Registration RSA | | | | | |
| imageRSA | User selected RSA image | String | Radio | Require | Y |
| questionRSA | User selected RSA question | String | Dropdown | Require | Y |
| answerRSA | Answer for select RSA question | String | Text | Require | Y |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbFTL** | | | | | |
| actionCode | Host integration action code | String | Host | N | Y |
| customerName | Customer name | String | Host | N | Y |
| customerType | Personal/Business | String | Host | N | Y |
| idNo | DOB for personal,  Business registration date for business | String | Host | N | Y |
| raceCode | Customer race | String | Host | N | Y |
| mobileNumber | Customer mobile number | String | Host | N | Y |
| emailAddress | Customer email address | String | Host | N | Y |
| accountList | Customer linked account list (vbAccount) | List | Host | N | N |
| rsaImageList | RSA image list for customer selection | List | RSA | Y | N |
| rsaQuestionList | RSA question list for customer selection | List | RSA | Y | N |
| status | 1st time log in status | String | IB | Y | Y |
| **vbAccount** | | | | | |
| acctType | Account type | String | Host | N | N |
| acctCode | Account code | String | Host | N | N |
| acctNo | Account number | String | Host | N | N |
| acctCurrency | Account currency | String | Host | N | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOPVRZZZ | Pin verification | 0001 0000 0000 |
| UOSETZZZ | Set IB 1st Time Logon Bit | 0002 0000 0000 |
| UOLGNCUS | Extract Customer Information | 0003 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_CREDENTIAL\_HISTORY | Registration history table | C |
| IBSS\_USER\_PROFILE | User profile table | C |
| IBSS\_ORGANIZATION | SME user only | C |
| \*\*UPass | Create user in UPass | C |

### Reset Password

**Module Overview**

This function allows user to reset password used in Internet Banking.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbFTL** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Reset Password Verification | | | | | |
| username | IB registered useraname | String | Text | Require  Alphanumeric  Min Length = 6  Max Length = 16 | - |
| identityNo | DOB for personal  Business registration date for busienss | String | Text | Require  Numeric | - |
| accountNo | Link account number | String | Text | Numeric | - |
| Reset Password Details | | | | | |
| password1 | Desire password | String | Password | Require  Alphanumeric  Min Length = 8  Max Length = 12 | Y |
| password2 | Reenter desire password | String | Password | Require  Alphanumeric  Min Length = 8  Max Length = 12 | N |
| rpc | Reset password code | String | Password | Require  Numeric  Length = 6 | N |
| Registration Result | | | | | |
| n/a |  |  |  |  |  |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbFTL** | | | | | |
| actionCode | Host integration action code | String | Host | N | Y |
| cifNo | CIF number | String | IB | N | Y |
| status | Reset password status | String | IB | Y | Y |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UORPCZZZ | Verify Details | 0011 0000 0000 |
| UORPCZZZ | RPC Verification | 0012 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_CREDENTIAL\_HISTORY | Registration/reset password history table | C |
| IB\_USER\_PROFILE | User profile table | R |
| \*\*UPass | Create user in UPass | U |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Login

**Module Overview**

Login module verifies user login details. Lock user upon max tries. With RSA image selected by user during 1st time login, user will be able to verify if the display image the selected by his/her.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbLogin** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Login Username | | | | | |
| username | IB registered useraname | String | Text | Require  Alphanumeric  Min Length = 6  Max Length = 16 | - |
| Login Password | | | | | |
| password | Desire password | String | Password | Require  Alphanumeric  Min Length = 8  Max Length = 12 | - |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbLogin** | | | | | |
| actionCode | Host integration action code | String | Host | N | N |
| cifNo | CIF number | String | IB | N | N |
| rsaImage | User selected rsaImage | String | IB | Y | N |
| accountList | User’s linked accounts list (vbAccount) | List | Host | N | N |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |
| **vbAccount** | | | | | |
| acctType | Account type | String | Host | N | N |
| acctCode | Account code | String | Host | N | N |
| acctNo | Account number | String | Host | N | N |
| acctCurrency | Account currency | String | Host | N | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOLGNCUS | Extract Customer Information | 0003 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_USER\_PROFILE | User profile table | R |
| \*\*UPass | Verify user in UPass | U |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Logout

**Module Overview**

Logout module allows user to logout from IB secure site. Logout summary will generate a logout summary containing activities performed by the user within the login session. Logout module will then invalidate the login session.

**Process Flow Overview**



**Form Bean**

Not available

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbLogout** | | | | | |
| logoutMessage | Logout general message | String | IB | Y | N |
| summaryList | Logout summary list | List | IB | Y | N |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |
| **vbSummary** | | | | | |
| trnxDate | Transaction date time | String | IB | Y | N |
| trnxDesc | Transaction description | String | IB | Y | N |
| trnxFromAcc | Transaction from account number | String | IB | Y | N |
| trnxToAcc | Transaction to account number | String | IB | Y | N |
| trnxAmount | Transaction amount | String | IB | Y | N |
| trnxRefNo | Transaction reference number | String | IB | Y | N |

**Integration Message Mapping**

Not available

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_TRANSACTION\_HISTORY | Internet banking transaction history table | R |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Error Logout

**Module Overview**

Error logout module is a general handler for invalidate user internet banking session in the event of system failure, invalid browser action detected and etc.

**Process Flow Overview**



**Form Bean**

Not available

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbErrorLogout** | | | | | |
| errorCode | Error code | String | Host/IB | Y | N |
| errorMessage | Error code related message | String | IB | Y | N |

**Integration Message Mapping**

Not available

**Database**

Not available

## Account Enquiry

### Account Summary

**Module Overview**

This function allows user to retrieve account summary for CASA accounts from MPI host. It carries summary details such like account type, balance and etc.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbAccountSummary** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Account Summary Page | | | | | |
| nextPage | Next page indicator | String | Link | - | - |
| View Bean | | | | | |
| viewBean | View bean for account summary module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbAccountSummary** | | | | | |
| currentPage | Current page number if available | String | IB | N | N |
| totalPage | Total page number if available | String | IB | N | N |
| summaryList | Account summary list | List | IB | Y | N |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |
| **vbAccount** | | | | | |
| acctType | Account type | String | Host | N | N |
| acctCode | Account code | String | Host | N | N |
| acctProdName | Account product name | String | IB | Y | N |
| acctNo | Account number | String | Host | Y | N |
| acctAvailBal | Account available balance | String | Host | Y | N |
| acctCurrBal | Account current balance | String | Host | Y | N |
| acctStatus | Account status | String | Host | Y | N |
| acctCurrency | Account currency | String | Host | Y | N |
| quickLinkMap | Available quick link list | Map | IB | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOASMZZZ | Account Summary of linkage accounts | 0021 0000 0000 |

**Database**

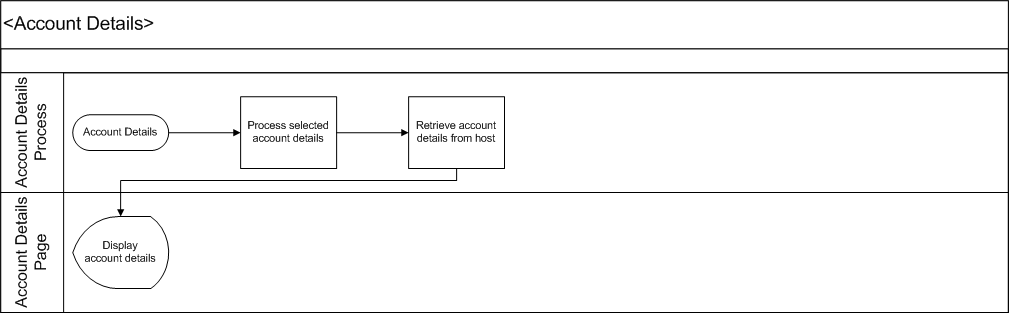
|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Account Details

**Module Overview**

Account details present more details on the selected account from account summary page.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbAccountDetails** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Account Details Process | | | | | |
| selectAcctNo | Selected account number | String | Link | - | - |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbAccountDetails** | | | | | |
| actionCode | Host integration action code | String | Host | N | N |
| currentDateMsg | Current date message | String | IB | Y | N |
| acctType | Account type | String | Host | Y | N |
| acctCode | Account code | String | Host | N | N |
| acctProdName | Account product name | String | IB | Y | N |
| acctNo | Account number | String | Host | Y | N |
| acctStatus | Account status | String | Host | Y | N |
| acctHolderNameList | Account holder name list | List | Host | Y | N |
| acctCurrBal | Account current balance | String | Host | Y | N |
| acctAvailBal | Account available balance | String | Host | Y | N |
| acctOneFloat | Account one day float amount | String | Host | Y | N |
| acctTwoFloat | Account two day float amount | String | Host | Y | N |
| acctOtherFloat | Account more than two day float amount | String | Host | Y | N |
| quickLinkMap | Available quick link list | Map | IB | Y | N |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOADTZZZ | Details of CA / SA Accounts | 0031 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Internet Transaction History

**Module Overview**

This feature allow customer to view transaction history of CASA account linked to the IB access performed at M2U.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbIBTrnxHist** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Reset Password Verification | | | | | |
| selectAcctNo | Selected account number | String | Link | - | - |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbIBTrnxHist** | | | | | |
| trnxHistList | Transaction history list | List | IB | Y | N |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |
| **vbTrnxHist** | | | | | |
| trnxDate | Transaction date time | String | IB | Y | N |
| trnxDesc | Transaction description | String | IB | Y | N |
| trnxFromAcc | Transaction from account number | String | IB | Y | N |
| trnxToAcc | Transaction to account number | String | IB | Y | N |
| trnxAmount | Transaction amount | String | IB | Y | N |
| trnxRefNo | Transaction reference number | String | IB | Y | N |

**Integration Message Mapping**

Not available

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_TRANSACTION\_HISTORY | IB transaction history table | R |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Transaction History

**Module Overview**

This feature allow customer to view transaction history of the CASA account linked to IB from current date up to 90 days. Transaction comprises of all activity performed from and to the account.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbAccountDetails** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Account Details Process | | | | | |
| selectAcctNo | Selected account number | String | Link | - | - |
| retrievalReferenceNo | Retrieval reference number if more records available | String | Link | - | - |
| todayIndicator | Indicator if desire option is today’s or past 90 days transaction | String | Link | - | - |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbAcctTrnxHist** | | | | | |
| actionCode | Host integration action code | String | Host | N | N |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |
| retrievalReference No | Retrieval reference for next page is more records available | String | Host | N | N |
| trnxHistList | Transaction history list (vbTrnxHistDetails) | List | Host | Y | N |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |
| **vbTrnxHistDetails** | | | | | |
| trnxDate | Transaction date | String | Host | Y | N |
| trnxDesc | Transaction description | String | Host | Y | N |
| trnxDebit | Transaction debit amount for debit transaction | String | Host | Y | N |
| trnxCredit | Transaction credit amount for credit transaction | String | Host | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOAIQZZZ | CA/SA today’s transaction history | 0071 0000 0000 |
| UOAIQZZZ | CA/SA 90 days transaction history | 0072 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IBSS\_SERVICE\_INFO | Service info content table | R |

## Cheque Management

### Reorder Cheque Book

**Module Overview**

This feature allows customer to request for cheque book via MPI’s IB.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbOrderCheque** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Cheque Book Reorder Details | | | | | |
| acctNo | Selected account number | String | Hidden | - | Y |
| noOfCheque | Number of cheque book to reorder | String | Text | Require  Numeric  Max length = ‘2’ | Y |
| typeOfCheque | Type of cheque book reordering  Individual  1.Personal  2.Commercial  Corporate  1) Corporate | String | Dropdown/  Label | Require | Y |
| collectionMode | Mode of collection for reorder cheque book | String | Dropdown/  Label | Require | Y |
| Cheque Book Reorder Confirmation | | | | | |
| n/a |  |  |  |  |  |
| Cheque Book Reorder Results | | | | | |
| n/a |  |  |  |  |  |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbOrderCheque** | | | | | |
| actionCode | Host action code | String | Host | N | N |
| acctNoDisp | Account number display | String | IB | Y | N |
| noOfChequeDisp | Number of cheque book display | String | IB | Y | N |
| typeOfChequeDisp | Type of cheque book display | String | IB | Y | N |
| collectionModeDisp | Collection mode display | String | IB | Y | N |
| trnxStatus | Transaction status | String | Host | Y | Y |
| trnxStatusMsg | Transaction status message | String | Host | N | N |
| trnxRefNo | Transaction reference no | String | IB | Y | Y |
| trnxDateTime | Transaction date time | String | IB | Y | Y |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOAIQZZZ | Reorder cheque book | 00H2 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_TRANSACTION\_HISTORY | IB transaction history table | C |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Cheque Warehousing

**Module Overview**

In this features user can perform enquiry on cheque(s) due for deposited to their account.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbChequeWarehouse** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Cheque Warehouse Details | | | | | |
| acctNo | Selected account number | String | Hidden | - | N |
| chequeDate | Post date for enquiry | String | Text | Require  Numeric  Match length = ‘8’ | N |
| Cheque Warehouse Summary | | | | | |
| retrievalRefNo | Retrieval reference number if more records available | String | Link | - | N |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbOrderCheque** | | | | | |
| actionCode | Host action code | String | Host | N | N |
| statusMessage | Status message if available | String | IB | Y | N |
| retrievalRefNo | Retrieval reference number if more records available | String | Host | N | N |
| chequeSummaryList | Cheque warehouse enquiry list | List | Host | Y | N |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |
| **vbChequeWarehouse** | | | | | |
| chequeDate | Post date | String | Host | Y | N |
| chequeNo | Cheque number | String | Host | Y | N |
| amount | Cheque amount | String | Host | Y | N |
| bankName | Drawee/issuer bank name | String | Host | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| - | Cheque warehousing enquiry | 00H5 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Cheque Status Enquiry / Stop Cheque

**Module Overview**

This feature allow customer to inquire the status of check that had been issued. Stop cheque feature is available for cheque status is UNPAID.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbChequeStatus** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Cheque Status Enquiry | | | | | |
| acctNo | Selected account number | String | Hidden | - | N |
| noOfCheque | No of cheque to be retrieved. Up to 20 cheques per enquiry | String | Dropdown | Require | N |
| chequeNo | Beginning cheque no for enquiry | String | Text | Require  Numeric | N |
| Cheque Status List | | | | | |
| selectedCheques | Selected cheque(s) to stop | String[] | Checkbox | Require | Y |
| Stop Cheque Confirmation | | | | | |
| stopReason | Reason for stop cheque | String | Text | Require  Max Length = ‘40’ | Y |
| Stop Cheque Results | | | | | |
| n/a |  |  |  |  |  |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbChequeStatus** | | | | | |
| actionCode | Host action code | String | Host | N | N |
| statusMessage | Status message if available | String | IB | Y | N |
| chequeStatusList | Cheque status enquiry list (vbChequeStatus) | List | Host | Y | N |
| stopChequesList | Selected stop cheques list (vbStopCheque) | List | User/IB | Y | Y |
| serviceInfo | Module content control service info bean | DynaBean | IB | Y | N |
| **vbChequeStatus** | | | | | |
| sequenceNo | Cheque display sequence number | String | IB | N | N |
| chequeNo | Cheque number | String | Host | Y | N |
| chequeStatus | Cheque status code | String | Host | N | N |
| chequeStatusDisp | Cheque status display | String | IB | Y | N |
| stopChequeReason | Stop cheque reason, only applicable for cheque status = ‘STOP’ | String | Host | Y | N |
| **vbStopCheque** | | | | | |
| sequenceNo | Sequence number for selected cheque | String | IB | N | N |
| chequeNo | Cheque number for selected | String | IB | Y | Y |
| referenceNo | Stop cheque reference number | String | IB | Y | Y |
| stopChequeStatus | Stop cheque result status cade | String | Host | N | N |
| stopChequeStatusDisp | String cheque result status display | String | IB | Y | Y |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOCSIZZZ | Enquiry status for a range of cheque numbers | 00H1 0000 0000 |
| - | Stop cheque | 00H3 0000 0000 |

**Database**

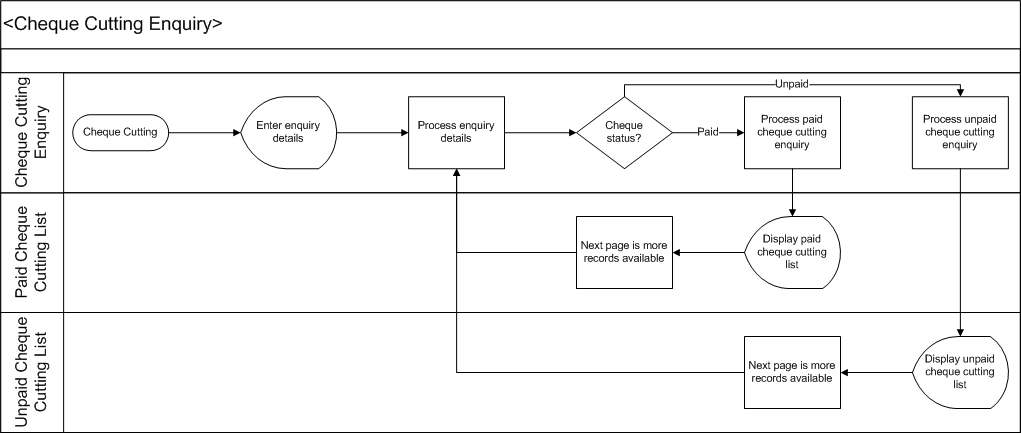
|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_TRANSACTION\_HISTORY | IB transaction history table | C |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Cheque Cutting Enquiry

**Module Overview**

This feature allows corporate customer to perform inquiry on cheque(s) submitted for check cutting to MPI.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbChequeCutting** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Cheque Status Enquiry | | | | | |
| acctNo | Selected account number | String | Hidden | - | N |
| chequeType | Type of cheque for enquiry  \*\* Corporate cheque | String | Dropdown/ Label | Require | N |
| chequeStatus | Cheque status for enquiry  \*\* Paid  \*\* Unpaid | String | Dropdown | Require | N |
| startCheque | Beginning cheque number | String | Text | Require  Numeric  Max Length = “12” | N |
| endCheque | End/last cheque number | String | Text | Require  Numeric  Max Length = “12”  >=startCheque | N |
| Paid Cheque Cutting List | | | | | |
| retrievalRefNo | Retrieval reference number if more records available (>30) | String | Text | Link | N |
| Un[aid Cheque Cutting List | | | | | |
| retrievalRefNo | Retrieval reference number if more records available (>30) | String | Text | Link | N |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbChequeCutting** | | | | | |
| actionCode | Host action code | String | Host | N | N |
| statusMessage | Status message if available | String | IB | Y | N |
| chequeCuttingList | Cheque cutting enquiry list (vbChequeCuttingDetails) | List | Host | Y | N |
| retrievalRefNo | Retrieval reference number if more records available (>30) | String | Host | N | N |
| serviceInfo | Service info view bean for module content control | DynaBean | IB | Y | N |
| **vbChequeCuttingDetails** | | | | | |
| chequeDate | Date of cheque Paid/Unpaid | String | Host | Y | N |
| chequeNo | Cheque number | String | Host | Y | N |
| amount | Cheque amount (Paid cheque only) | String | Host | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOCCSZZZ | Cheque cutting enquiry | 00H4 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IBSS\_SERVICE\_INFO | Service info content table | R |

## Account Maintenance

### Link Account

**Module Overview**

This feature allows customer to link his/her bank account for IB access.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbLinkAccount** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Link Account Details | | | | | |
| acctNo | Selected account number | String | Text | Require  Numeric | Y |
| Link Account Confirmation | | | | | |
| tac | TAC number | String | Password | Require  Numeric  Match Length = “6” | N |
| Cheque Book Reorder Results | | | | | |
| n/a |  |  |  |  |  |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbLinkAccount** | | | | | |
| actionCode | Host action code | String | Host | N | N |
| acctType | Account type | String | Host | N | N |
| productCode | Number of cheque book display | String | Host | N | N |
| productName | Account product name | String | IB | N | Y |
| acctCurrency | Type of cheque book display | String | Host | N | Y |
| trnxStatus | Transaction status | String | Host | Y | Y |
| trnxStatusMsg | Transaction status message | String | Host | Y | N |
| trnxRefNo | Transaction reference no | String | IB | Y | Y |
| trnxDateTime | Transaction date time | String | IB | Y | Y |
| serviceInfo | Service info view bean for module content control | DynaBean | IB | Y | N |
| serviceInfo | Service info view bean for module content control | DynaBean | IB | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| - | Link account | 0051 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_TRANSACTION\_HISTORY | IB transaction history table | C |
| IBSS\_SERVICE\_INFO | Service info content table | R |

### Unlink Account

**Module Overview**

This feature allows customer to unlink his/her bank account from IB access.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbLinkAccount** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Link Account Details | | | | | |
| acctNo | Selected account number | String | Text | Require  Numeric | Y |
| Link Account Confirmation | | | | | |
| tac | TAC number | String | Password | Require  Numeric  Match Length = “6” | N |
| Cheque Book Reorder Results | | | | | |
| n/a |  |  |  |  |  |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbLinkAccount** | | | | | |
| actionCode | Host action code | String | Host | N | N |
| acctType | Account type | String | IB | N | N |
| productCode | Number of cheque book display | String | Host | N | N |
| productName | Account product name | String | IB | N | Y |
| acctCurrency | Type of cheque book display | String | Host | N | Y |
| trnxStatus | Transaction status | String | Host | Y | Y |
| trnxStatusMsg | Transaction status message | String | Host | Y | N |
| trnxRefNo | Transaction reference no | String | IB | Y | Y |
| trnxDateTime | Transaction date time | String | IB | Y | Y |
| serviceInfo | Service info view bean for module content control | DynaBean | IB | Y | N |
| serviceInfo | Service info view bean for module content control | DynaBean | IB | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| - | Link account | 0052 0000 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_TRANSACTION\_HISTORY | IB transaction history table | C |
| IBSS\_SERVICE\_INFO | Service info content table | R |

## Fund Transfer

### Own Account Transfer

**Module Overview**

This feature allows customer to perform funds transfer to their own linked account number via MPI’s IB.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbOwnTransfer** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| Own Account Transfer Details | | | | | |
| fromAcct | Selected from account number | String | Dropdown/ Label | Require | Y |
| toAcct | Selected to account number | String | Dropdown | Require | Y |
| amount | Transfer amount | String | Text | Require  Numeric | Y |
| effectiveDate | Date for transfer instruction | String | Dropdown | Require | Y |
| Own Account Transfer Confirmation | | | | | |
| n/a |  |  |  |  |  |
| Own Account Transfer Results | | | | | |
| n/a |  |  |  |  |  |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbOwnTransfer** | | | | | |
| actionCode | Host action code | String | Host | N | N |
| fromAcctMap | From account number list | Map | IB | Y | N |
| toAcctMap | To account number list | Map | IB | Y | N |
| effectiveDateMap | Effective date selection list | Map | IB | Y | N |
| fromAcctDisplay | From account info display | String | IB | Y | N |
| fromAcctBal | From account balance after transfer | String | Host | N | Y |
| fromAcctBalDis | From account balance display | String | IB | Y | N |
| toAcctDisplay | To account info display | String | IB | Y | N |
| toAcctBal | To account balance after transfer | String | Host | N | Y |
| toAcctBalDis | To account balance display | String | IB | Y | N |
| amountDis | Transfer amount display | String | IB | Y | N |
| effectiveDateDis | Effective date display | String | IB | Y | N |
| trnxStatus | Transaction status | String | Host | Y | Y |
| trnxStatusMsg | Transaction status message | String | Host | Y | N |
| trnxRefNo | Transaction reference no | String | IB | Y | Y |
| trnxDateTime | Transaction date time | String | IB | Y | Y |
| serviceInfo | Service info view bean for module content control | DynaBean | IB | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOOLPZZZ | Immediate / Future own account transfer | 00E1 xxxx xxxx |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_TRANSACTION\_HISTORY | IB transaction history table | C |
| IBSS\_SERVICE\_INFO | Service info content table | R |
| IBSS\_PRODUCT\_LIST | Product list content table | R |
| IBSS\_PRODUCT\_INFO | Product info content table | R |

### 3rd Party Account Transfer

**Module Overview**

This feature allows customer to perform funds transfer to another 3rd party MPI’s peso account via MPI’s IB.

**Process Flow Overview**



**Form Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **fbIntraTransfer** | | | | | |
| **Property** | **Description** | **Type** | **Input Type** | **Validation** | **DB Entry** |
| 3rd Party Transfer Details | | | | | |
| fromAcct | Selected from account number | String | Dropdown/ Label | Require | Y |
| toAcct | Selected to account number from favorite list or enter account number | String | Dropdown/Text | Require  Numeric  Max Length = “16” | Y |
| amount | Transfer amount | String | Text | Require  Numeric | Y |
| effectiveDate | Date for transfer instruction | String | Dropdown | Require | Y |
| 3rd Party Transfer Confirmation | | | | | |
| tac | TAC value | String | Password | Require  Numeric  Match Length = “6” | N |
| 3rd Party Transfer Results | | | | | |
| n/a |  |  |  |  |  |
| View Bean | | | | | |
| viewBean | View bean for FTL module | DynaBean | - | - | - |

**View Bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Description** | **Type** | **Source** | **Display** | **DB Entry** |
| **vbIntraTransfer** | | | | | |
| actionCode | Host action code | String | Host | N | N |
| fromAcctMap | From account number list | Map | IB | Y | N |
| favAcctMap | Favourite to account number list | Map | IB | Y | N |
| effectiveDateMap | Effective date selection list | Map | IB | Y | N |
| toAcctHoldername | To account holder name | String | Host | Y | N |
| fromAcctDisplay | From account info display | String | IB | Y | N |
| fromAcctBal | From account balance after transfer | String | Host | N | Y |
| fromAcctBalDis | From account balance display | String | IB | Y | N |
| amountDis | Transfer amount display | String | IB | Y | N |
| effectiveDateDis | Effective date display | String | IB | Y | N |
| trnxStatus | Transaction status | String | Host | Y | Y |
| trnxStatusMsg | Transaction status message | String | Host | Y | N |
| trnxRefNo | Transaction reference no | String | IB | Y | Y |
| trnxDateTime | Transaction date time | String | IB | Y | Y |
| serviceInfo | Service info view bean for module content control | DynaBean | IB | Y | N |

**Integration Message Mapping**

|  |  |  |
| --- | --- | --- |
| **Message ID** | **Description** | **Service Code** |
| UOOLPZZZ | Immediate / Future 3rd party account transfer | 00F1 xxxx 0000 |

**Database**

|  |  |  |
| --- | --- | --- |
| Table | Description | CRUD |
| IB\_TRANSACTION\_HISTORY | IB transaction history table | C |
| IB\_INTRA\_TRANSACTION | IB 3rd party transaction history table | C |
| IBSS\_SERVICE\_INFO | Service info content table | R |
| IBSS\_PRODUCT\_LIST | Product list content table | R |
| IBSS\_PRODUCT\_INFO | Product info content table | R |

### Interbank Account Transfer

### Favorite Maintenance

### Future Transfer

### Cash Card Top Up