# Release Bulletin Sample

(Suite Name) (Product Name) **- Automate TMPG Reporting**

|  |  |
| --- | --- |
| **Enhancement #** | XXXXXXX |
| **Business Objective** | (Suite Name) to automate the reporting of failed transactions as prescribed by Treasury Market Practice Group  (TMPG). |
| **Application Affected** | (Suite Name) (Product Name) |
| **Description** | (Suite Name) is enhanced as follows:  1. Identifies the failed transaction(s) based on the following condition:  clearance date > settlement date.  2. Derives the fail charges based on the following formula:  Fail Charges = (TMPG reference rate \* Failing Settlement Amount \* Number of days failing)/360   * TMPG reference rates are maintained in the application. * If a trade fails on day 1, then the **Fail Days** column is updated as ‘**1**’. For subsequent fails, the **Fail Days**   column is updated accordingly and fail charges is calculated.   * Even weekends and trading holidays are considered as active fail days.   3. Reports the failed transactions and their fail status on the **CTA** screen under the three columns **Fail**  **Charges**, **Fail Days**, and **Claim Status**.  You can add your reason(s) for fails under the **Comment** column.   * The manual actions **Waive off** and **Close** (entitlement controlled) are enabled on the **CTA** screen and **Multi Action** drop-down for these failed transactions.   4. Purges these failed transactions (which are either in **Settled** or **Waive off** status) from the application after a  number of days defined in the application.  These failed transaction(s) can be audited from the **View Details** window. |
| **Business Benefit** | (Suite Name) automatically calculates the fail charges and reports their transaction(s) on its UI, thus helping clients avoid calculate fail charges manually and view those transaction(s). |
| **Screen Modification** | The following screenshot shows the below listed items on the **CTA** screen:  l **Fail Days**, **Fail Charges**, and **Claim Status** fields containing values  l Manual Actions–**Close**, **Revise Counter Party**, and **Waive off**.  The following screenshot shows the manual actions **Waive off** and **Close** options on the **Choose Action** dropdown:  of the **Multi Actions** window: |
| **Additional Setup** | Not Applicable |
| **Limitations** | * Settlement of receipt/deliver fail charges is handled outside the application. * If failed transaction(s) gets either closed/waived off, then no updates are sent to the client source system. * For subsequent transaction(s) fails, (Suite Name) does not show the previous fail charges. * Only two sets of fail reference rates are available to maintain in the application at any given point in time. * For failed transactions on which cancel/amendment have been received, (Suite Name) does not calculate the fail charges. |
| **Related Enhancements** | Not Applicable |
| **More Details** | Not Applicable |

# Release Fix Report (Enhancement) Sample

|  |  |
| --- | --- |
| Enhancement Number | XXXXXX |
| Severity | Medium |
| Summary | (Suite Name) receives the SWIFT pledge instruction message from the client source system and proceeds as follows:   * Populates the Bank Loan Account Number field with the default value of "000000002378" on the BPS-CAGE pledge inbound message. This field value is no longer mapped and populated from the 97A tag.   The default value can be configured from the DB script.   * Populates the Customer Account Number field on the BPS-CAGE pledge inbound message based on the value populated on the 97A tag as follows:   + If the 97A tag has the value populated with fewer than 20 characters such as "123456789A," then this field is padded with empty spaces to fill 20 characters and populated as " 123456789A."      * + If the 97A tag has the value populated with more than 20 characters such as 123456789ABCDEFGHIJKLMNO, then this field is populated by deleting the final 4 characters such as "123456789ABCDEFGHIJK." |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| Test Procedure | Test Step | Expected Result |
|  |  |  |
| 1. Validate the 'Bank Loan Account Number' value populated in Pledge inbound message for MT540 | | |
|  | 1. Process OCC pledge incoming SWIFT instruction– MT540  2. Login to the application and Navigate to CTA  3. Find the trade in CTA and right click it, click view details  4. Go to audit and click show message-->Raw Message for (Product Name) message  5. Validate the message for "Bank Loan Account Number" value to be defaulted to ‘000000002378’, at the position 77 to 88 (12 char fixed length) | 1. OCC pledge SWIFT instruction must be processed successfully to Clearance Pro  2. User must be logged in successfully and CTA page must be displayed  3. View details pop must be displayed  4. (Product Name) Raw Message must be displayed  5. ‘000000002378' 12 char fixed length number must be defaulted for Bank loan account number from 77 to 88 positions |
| 2. Validate the 'Bank Loan Account Number' value populated in Pledge inbound message for MT542 | | |
|  | 1. Process OCC pledge incoming SWIFT instruction– MT542  2. Login to the application and Navigate to CTA  3. Find the trade in CTA and right click it, click view details  4. Go to audit and click show message-->Raw Message for (Product Name) message  5. Validate the message for "Bank Loan Account Number" value to be defaulted to ‘000000002378’, at the position 77 to 88 (12 char fixed length) | 1. OCC pledge SWIFT instruction must be processed successfully to (Suite Name)  2. User must be logged in successfully and CTA page must be displayed  3. View details pop must be displayed  4. (Product Name) Raw Message must be displayed  5. ‘000000002378' 12 char fixed length number must be defaulted for Bank loan account number from 77 to 88 positions |
|  |  |  |

# Release Fix Report (Defect) Sample

Defect Number (XXXXXXX)

Details

|  |  |
| --- | --- |
| Application Affected | XXX |
| Severity | High |
| Summary | The PairOff Confirmation dialog box did not display the parent and resultant trades. |
| Details | After the Pairoff manual action on the CTA screen for a trade having an External Status as Pending, the PairOff Confirmation dialog box displayed just the Loading… text on it but displayed neither the resultant nor the parent trades even after a long wait.  Further, the parent trades were displayed in Pending status, but the resultant trade was not displayed on the CTA screen. |
| Solution | The parent trades and resultant trade display on the PairOff Confirmation dialog box and CTA screen. |

Test Procedure

|  |  |  |
| --- | --- | --- |
| Step No. | Description | Expected Result |
| 1. | Log on to ClearancePro application. | User must be logged in successfully. |
| 2. | Navigate to Views>Position View screen | User must be able to navigate to the Position View screen successfully. |
| 3. | Validate the Position View screen. | There must be numerous line items with different securities and their positions in different accounts. |
| 4. | Perform a manual action on the one of the line items (CA135087VH40). | All the below manual options must be displayed.   * Select All * Select None * Inter Account Movement * New South Bound Trade * New Trade * Position Change Details * Show All Outstanding Activity |
| 5. | Click the Inter Account Movement manual option. | Inter Account Movement screen must be successfully opened. |
| 6. | Validate the Acc Details. | Acc Details must be "SA000 and GA000." |

# User Guide (Sample)

Setup

The administrator has privileges to set up accounts in the application.

From the application menu, select Setup. The following menu items are displayed:

* Users
* Organizations
* Groups
* Entitlements
* SOD Cap Set up
* Users Login History
* Scheduler Notification
* Potential Matching
* Alert Notification
* Match Rules ICM
* Security Maturity Alerts
* Customer Mapping Configuration

**Organizations Screen**

As an administrator, you have the privilege to set up the organization details of the user in the application.

## Accessing the Organizations Screen

From the application menu, select Setup > Organizations. The Organizations screen displays. [See Organizations Screen: Fields and Descriptions on the next page.](XXX)

## Viewing the Organizations List

You can view all organizations available in the application.

To view organizations list:

1. Access the Organizations screen.

2. Click the icon.

3. Enter the search criteria, then click the icon. The Organizations screen displays all organizations of various users available in the application.

## Adding an Organization Information

You can add an organization information to the application.

To add an organization information:

1. Access the [Organizations screen](XXXX).

2. Click the icon. The Add Organization Details window displays to enter the details.

3. Make the necessary changes.

4. Click Add. The organization with its details is added to the application.

## Modifying an Organization Information

You can modify an organization information in the application.

To modify an organization information:

1. Access the [Organizations screen](XXXX).

2. Click the icon. The Edit Organization Details window displays to edit the details.

3. Make the necessary changes.

4. Click Update. The edited details are updated in the application.

## Deleting an Organization Information

You can delete an organization information in the application.

To delete an organization information:

1. Access the [Organizations screen](XXXXX).

2. Click the icon. The delete confirmation window displays asking whether you want to delete the organization.

3. Click OK. The organization with its details is deleted from the application.

## Organizations Screen: Fields and Descriptions

|  |  |
| --- | --- |
| Field | Description |
| Icon |  |
| Icon-1 | Description |
| Icon-2 | Description |
| Icon-3 | Description |
| **Organizations** |  |
| Organization Name | Name of the organization |
| Address | Address of the organization |
| Layout Name | Layout name of the organization |
| Style Name | Style name of the organization |

Code Documentation using “Visual Studio Code” editor and generating the document using the automation generation tool (JSDoc)

Index.js file

// @ts-check

const { caltotal } = require("./petcounter");

/\*\*

 \* @file index.js is the homepage for this application

 \* @author Mithun

 \* @see <a href="https://en.wikipedia.org/wiki/India">India</a>

 \*/

/\*\*

 \* Pet name

 \* @type {string}

 \*/

const string = "Ruby";

/\*\*

 \* Pet number

 \* @type {number}

 \*/

const number = 100;

/\*\*

 \* My Array

 \* @type {Array<number>}

 \*/

const myArray\_number = [10, 132.12, 100];

// const myArray = [10, 132.12, 100, "Hi", true]

/\*\*

 \* Pet object

 \* @type {{id: number, name: string, age: number|string}}

 \*/

const object = {

  id: 1,

  name: "Coco",

  age: "2",

};

/\*\*

 \* Calculate pet age

 \* @param {number} current current year

 \* @param {number} yearOfBirth year of pet birth

 \* @returns {string} pet age

 \*/

const calculateAge = (current, yearOfBirth) => {

  return `${current - yearOfBirth}`;

};

console.log(calculateAge(2021, 2019));

//////////////////////////

/\*\*

 \*

 \* @typedef {Object} Dog

 \* @property {number} id

 \* @property {string} name

 \* @property {number|string} age

 \* @property {boolean} [isMale] gender {optional}

 \*/

/\*\*

 \* Custom Object

 \* @type {Dog}

 \*/

const custom\_object = {

  id: 1,

  name: "Bean",

  age: 2,

  // isMale: true,

};

/\*\*

 \* Class to create a new pet owner

 \*/

class Owner {

  /\*\*

   \* Pet owner detail

   \* @param {Object} ownerDetail

   \*/

  constructor(ownerDetail) {

    /\*\*

     \* @property {string} name pet owner name

     \*/

    this.name = ownerDetail.name;

    /\*\*

     \* @property {number} age pet owner age

     \*/

    this.age = ownerDetail.age;

  }

  /\*\*

   \* @property {Function} printOwner print out owner information

   \* @returns {void}

   \*/

  printOwner() {

    console.log(`Owner's name is ${this.name} and her age is ${this.age}`);

  }

}

/\*\*

 \* Link to Owner class

 \* See {@link Owner}

 \*/

const owner\_call = new Owner({

  name: "Kelly",

  age: 18,

});

owner\_call.printOwner();

console.log(caltotal(19, 5));

**Output:**

