BNY **Mellon**

Detail Design Document

**PSS – Transaction Console**

Author: Prabhu VSD & Amit Saini

**Document History** – describe the changes to this document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Date** | **Entered by** | **Comments** |
| 0.01 | 06/24/2021 | Prabhu VSD |  |
| 0.01 | 06/24/2021 | Amit Saini |  |
|  |  |  |  |

**Table of Contents**

[i. Document Approvals 3](#_Toc64479901)

[1 Document Purpose 4](#_Toc64479902)

[2 Background 4](#_Toc64479903)

[2.1 Executive Summary 4](#_Toc64479904)

[3 System Number. 4](#_Toc64479905)

[3.1 System Name 4](#_Toc64479906)

[3.2 System Description 4](#_Toc64479907)

[3.3 Architecture Overview 4](#_Toc64479908)

[3.4 Data Details 21](#_Toc64479909)

[3.4.1 Physical Database Design 22](#_Toc64479910)

[3.4.2 Archive and Purge consideration 22](#_Toc64479911)

[3.4.3 Backup and recovery strategy 22](#_Toc64479912)

[3.5 Hardware Details 22](#_Toc64479913)

[3.6 Middleware Details 22](#_Toc64479914)

[3.7 Detailed Design 22](#_Toc64479915)

[3.7.1 Release Reversal Entitlement Breaks Task 22](#_Toc64479916)

[3.7.2 Existing Power Builder Screen Details 24](#_Toc64479917)

[3.7.3 Income MOD changes 41](#_Toc64479918)

[3.7.4 Technical Design solution 42](#_Toc64479919)

[3.8 Software Details 43](#_Toc64479920)

[3.8.1 Error Handling 43](#_Toc64479921)

[3.9 System Interface Details 43](#_Toc64479922)

[3.10 Network Details 44](#_Toc64479923)

[4 Code Quality Reviews 44](#_Toc64479924)

[5 Logging Policy 44](#_Toc64479925)

[6 Implementation Considerations 44](#_Toc64479926)

[7 Contributors 45](#_Toc64479927)

[8 Terms and Abbreviations 46](#_Toc64479928)

[9 Support and Reference Documents 47](#_Toc64479929)

[10 Appendix A – On Line Specifications 48](#_Toc64479930)

[11 Appendix B – Report Specifications 49](#_Toc64479931)

1. Document Approvals

The stakeholders listed below need to approve this document. When stakeholders approve this document, they are indicating that they have carefully:

* Read;
* Reviewed;
* Considered the impact of this document to their organization;
* And agree that the document is:
* Accurate
* Complete
* Sufficient
* Consistent with the project needs
* Meets BNY Mellon standards and best practices

Approval documentation must be kept on file in the project repository.

Signoff matrix design document:

A= Approver

R= Reviewer

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | [**ISTPSS-8805**](https://jira14.bnymellon.net/browse/ISTPSS-8805) **and** [**ISTPSS-9300**](https://jira14.bnymellon.net/browse/ISTPSS-9300)  **Reversal Entitlement Break Queue**  **Task Creation Income**  **Resolve Tasks GCE and Income** | **Signoff status** |
| Manas Wagh | GPO Client Inquiry | R |  |
| Ellie Blondeel | GPO Voluntary Corporate Actions | A |  |
| Mike Skolmowski | GPO Mandatory Corporate Actions | A |  |
| Yogesh Jawalkar/Priyanka Labh | GPO Event Creation | R |  |
| Ravi Kumar | GPO High Profile | A |  |
| Michael Rakowski/ | GPO Lending | R/A |  |
| Basha Shaik |  |
| Jonathan Waldinger/ | GPO GBS | A |  |
| Chris Toye |  |
| Loveen Motwani/Matt Johnson | GPO Post Payment | A |  |
| Harshad Nirgudkar | GPO Income | A |  |
| Maureen Lake | GPO Governance & Control | R |  |

GPO Lending: approver if lending related tasks

GPO Post Payment: approver if post-payment related tasks

<<Need to attach signoff emails>>

# Document Purpose

The purpose of this document is to present the Detailed Design Document for Resolving tasks for transaction console entitlements reversal using current NEXEN workflow.

# Background

Using the user specified inputs the records pertaining to transaction were retrieved and populated in the landing page.

## Executive Summary

User Story – Tasks for Transaction Console

<https://jira14.bnymellon.net/browse/CTSNXNPP-15524>

# System Number.

@Arvind will give it

## System Name

PSS

## System Description

PSS is the processing engine at BNYM that allows the event processing of corporate action events and tax reclaims. It calculates client entitlements and produces downstream cash/securities movements. The application automates the process which reduces costs and reduces risk of errors. The application is a critical feeder (direct or indirect) to an active financial processing application. PSS also process Income and Tax Events. It provides reclaims processing capabilities that are applicable in the market. It also provides for STP processing for Call Bond Processing in addition to processing maturities in all markets Application is used by Custody, middle -office outsourcing within CSD. The Application contains Highly Confidential Information (HCI) data.

## Architecture Overview

### Transaction Console Landing Page

This is the main page which mainly consists of Filter Component. After applying Filters, corresponding Transaction results are shown through Statistics Component as well as Transaction Grid Component.

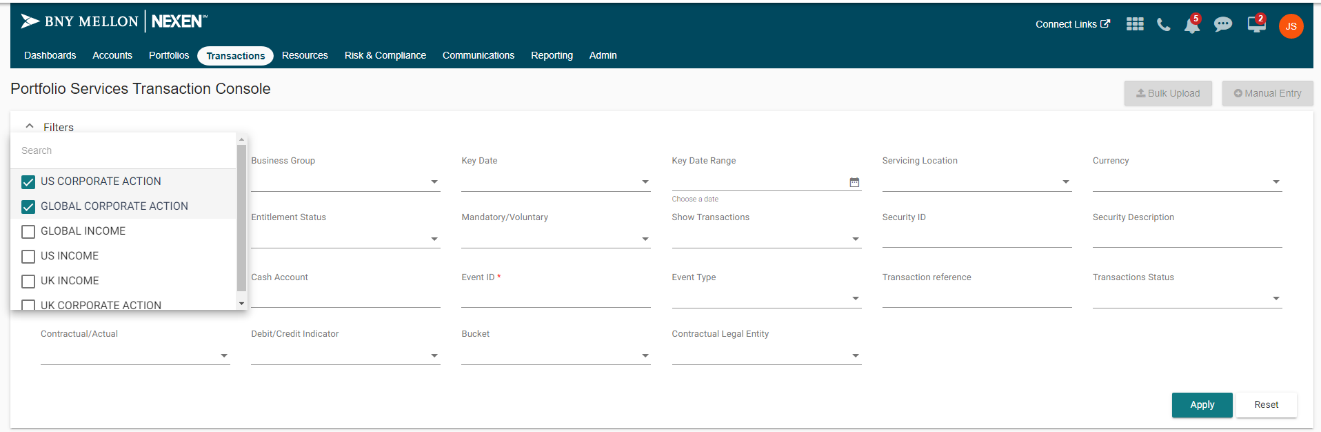
#### Filter Components

We have 22 filter inside the filter component on transaction console page . when we apply the filter the output comes in the grid view on the same page .

Following are the filters input :-

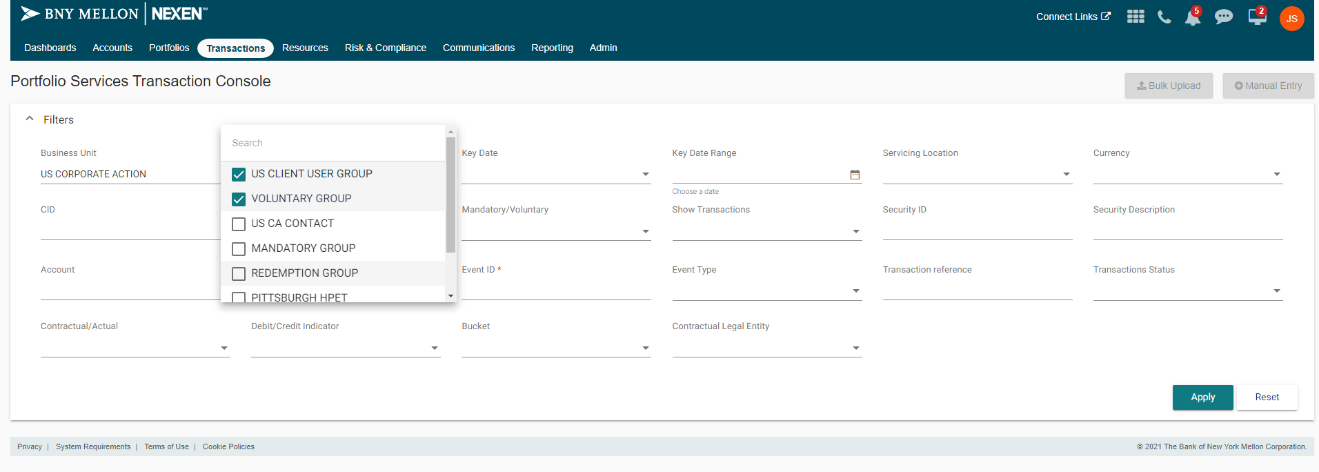
**1. Business Unit** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list. It’s value is dependant on Business Group filed value.



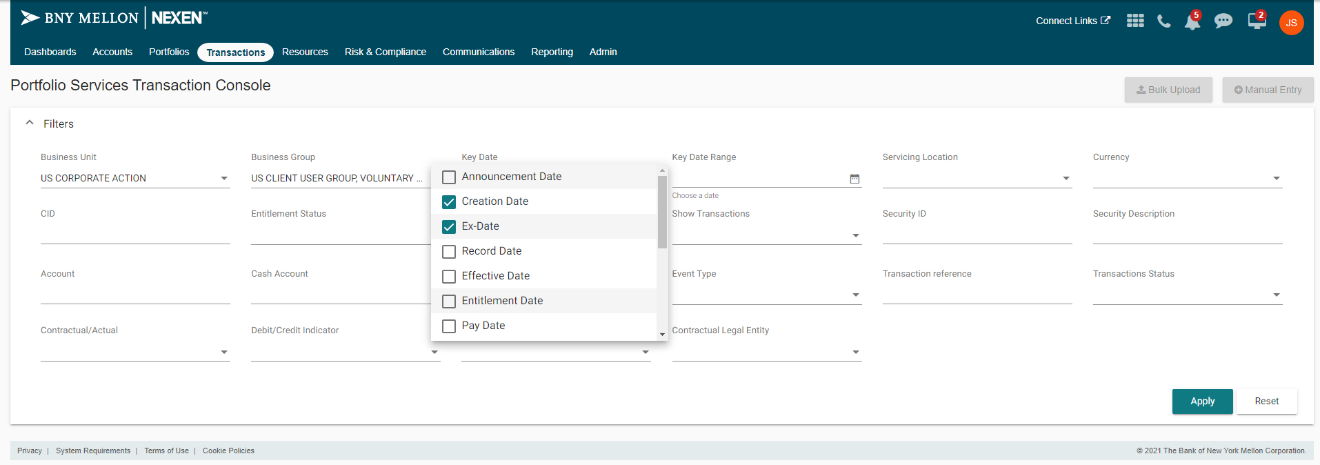
**2. Business Group** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list. It’s value is dependant on Business Unit filed value .



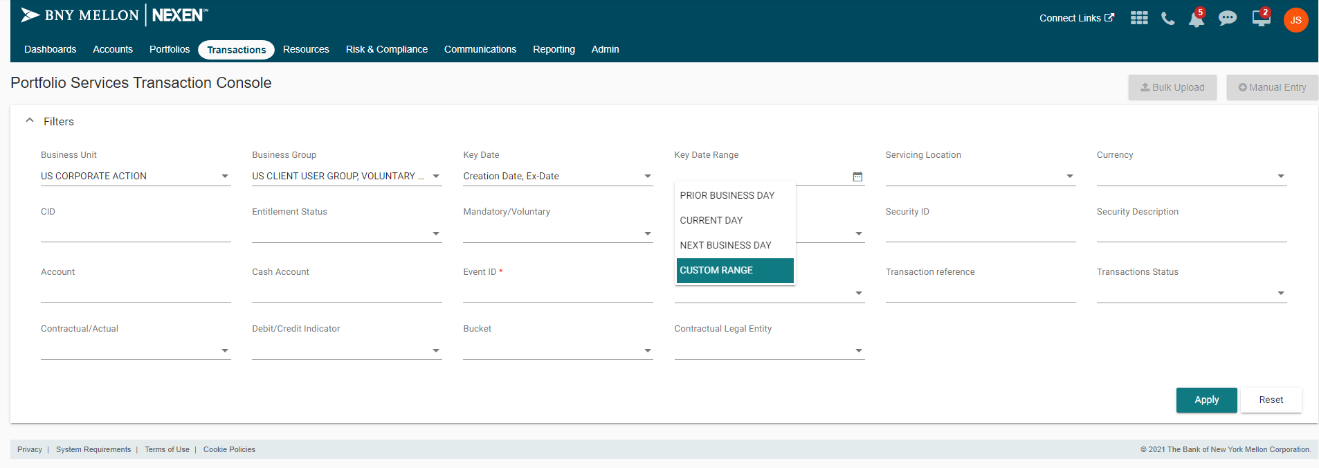
**3.Key Date** dropdown selection

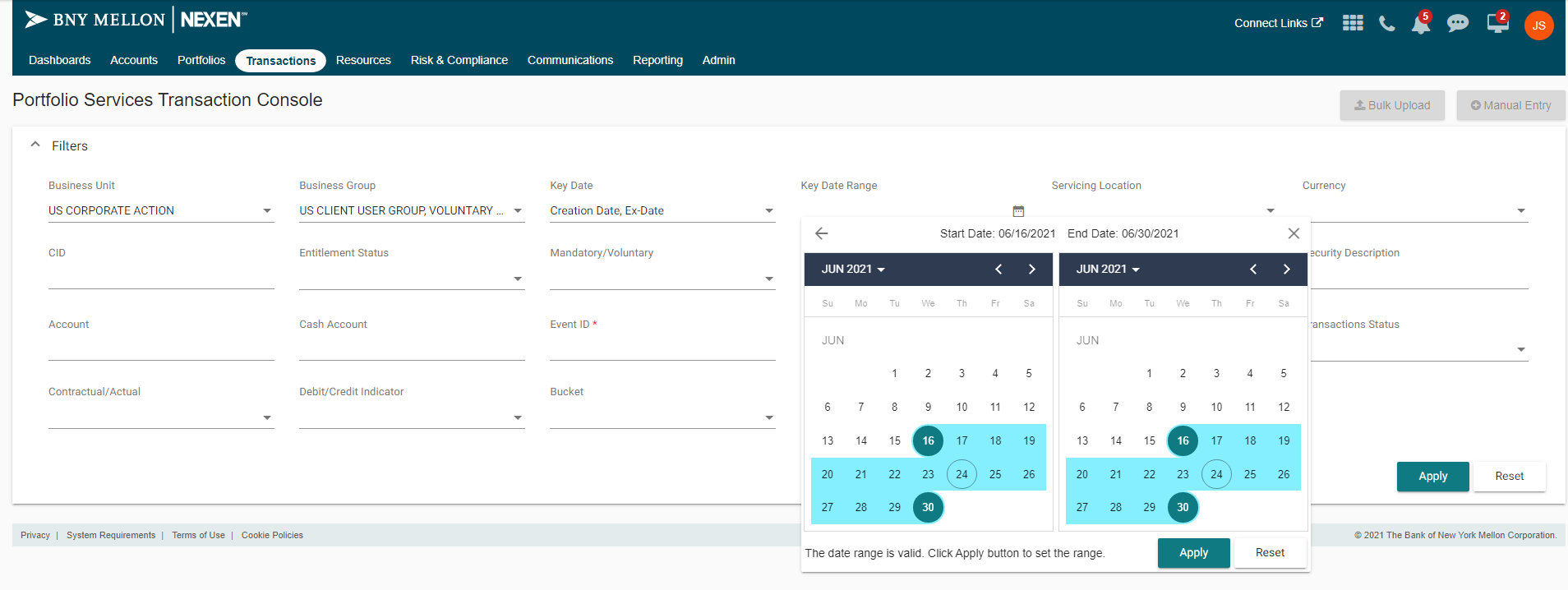
It is multi valued dropdown, user can select the multiple values from dropdown list.



4.**Key Date Range** dropdown selection

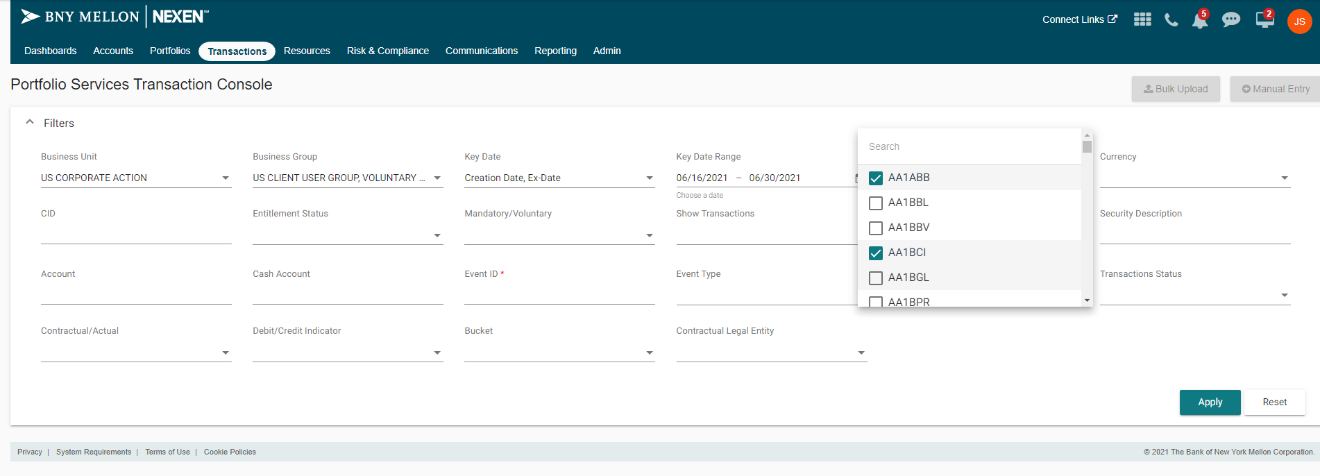
It is multi valued dropdown, user can select the multiple values from dropdown list.





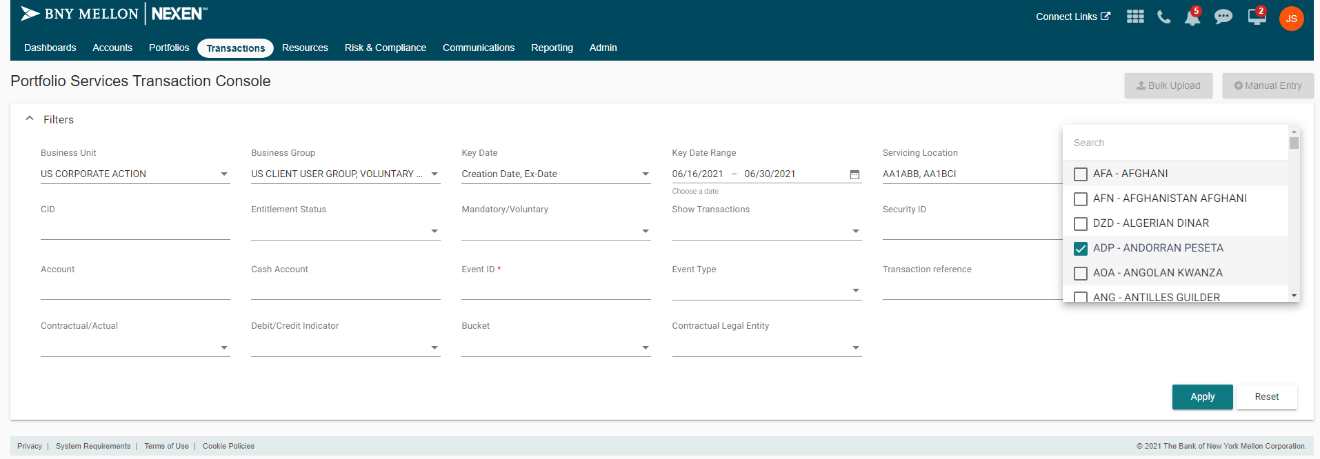
**5. Servicing Location** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list.



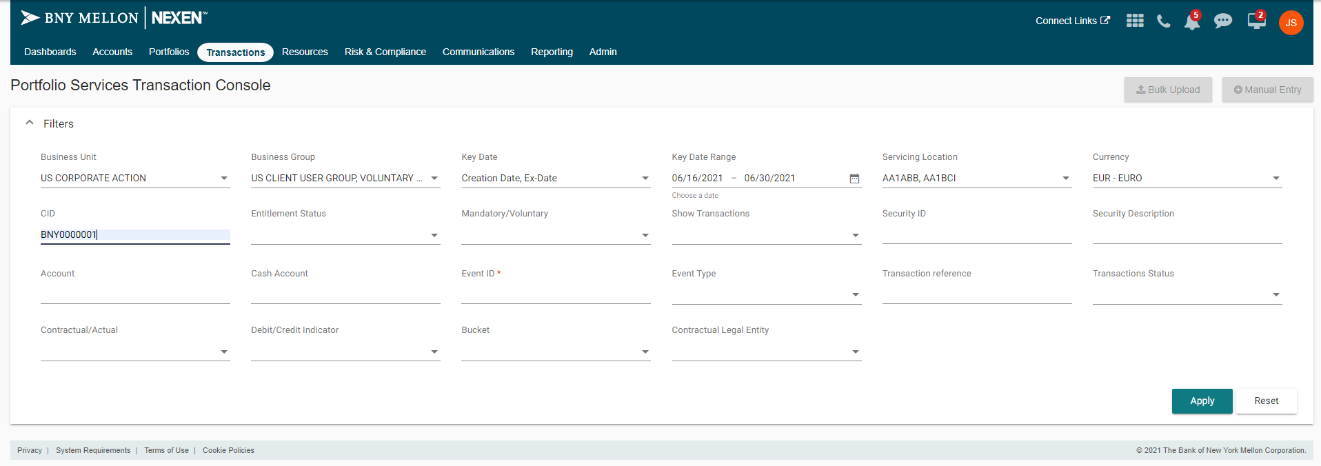
**6. Currency dropdown selection**

It is multi valued dropdown, user can select the multiple values from dropdown list.



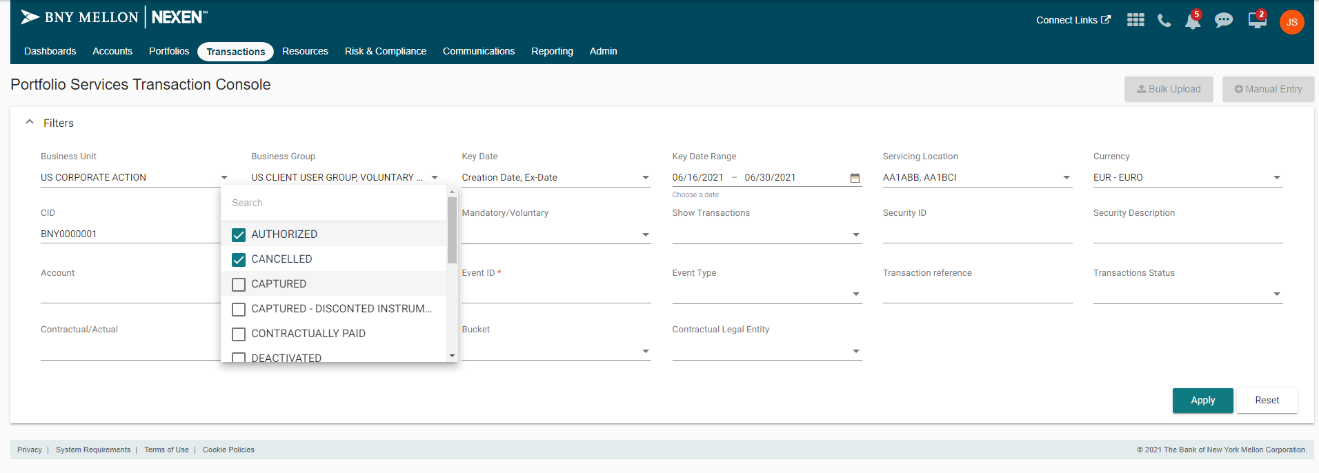
**7. CID** Input Field

It is a Input field which allows user to enter only Alpha Numeric values for this field.



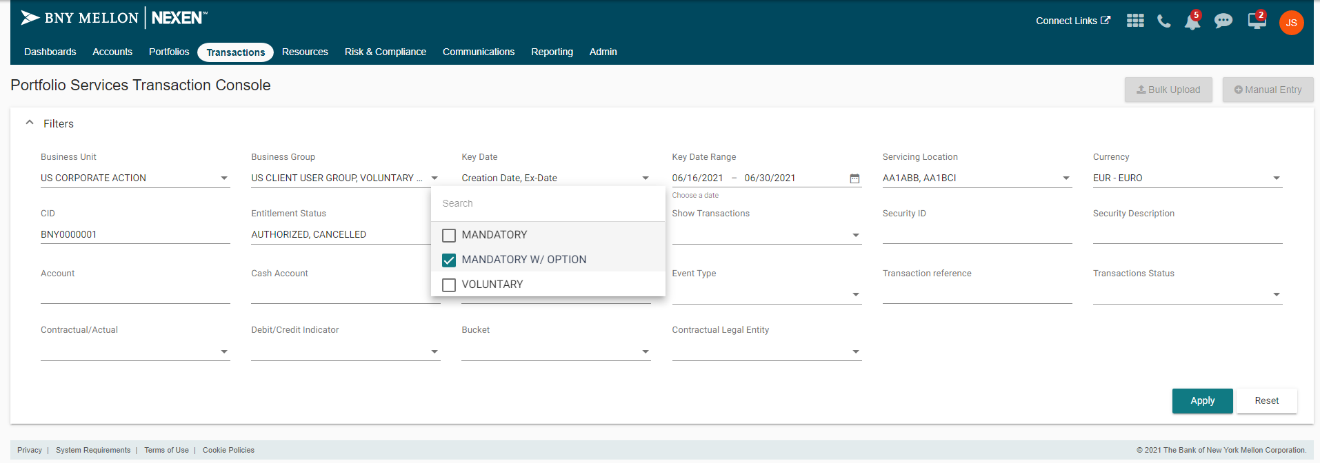
**8. Entitlement Status** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list.



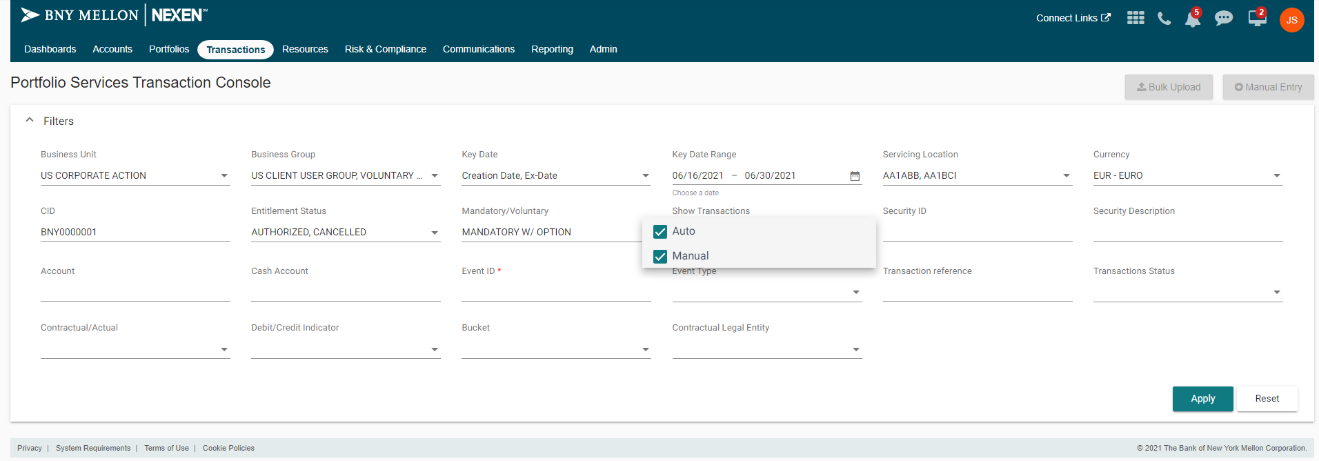
**9. Mandatory/Voluntary** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list



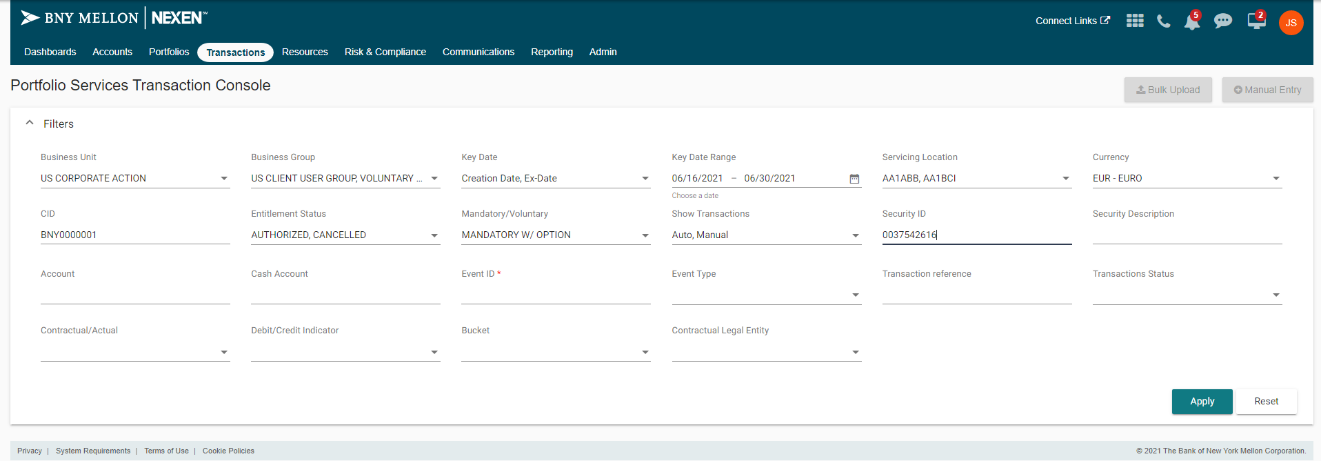
**10. Show Transactions** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list



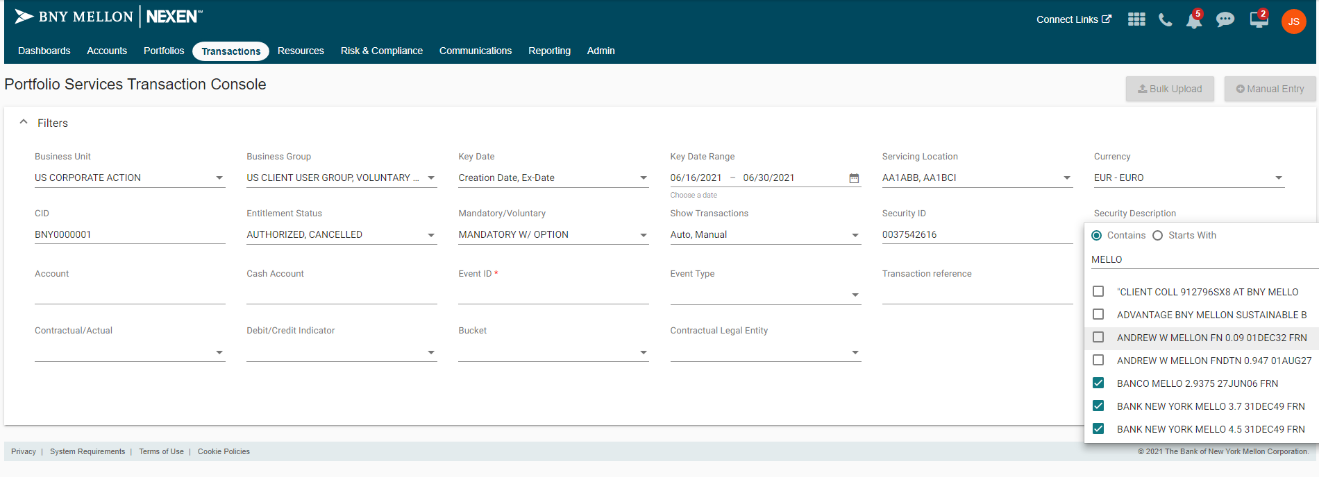
**11. Security ID** Input Field

It is a Input field which allows user to enter only Numeric values for this field.



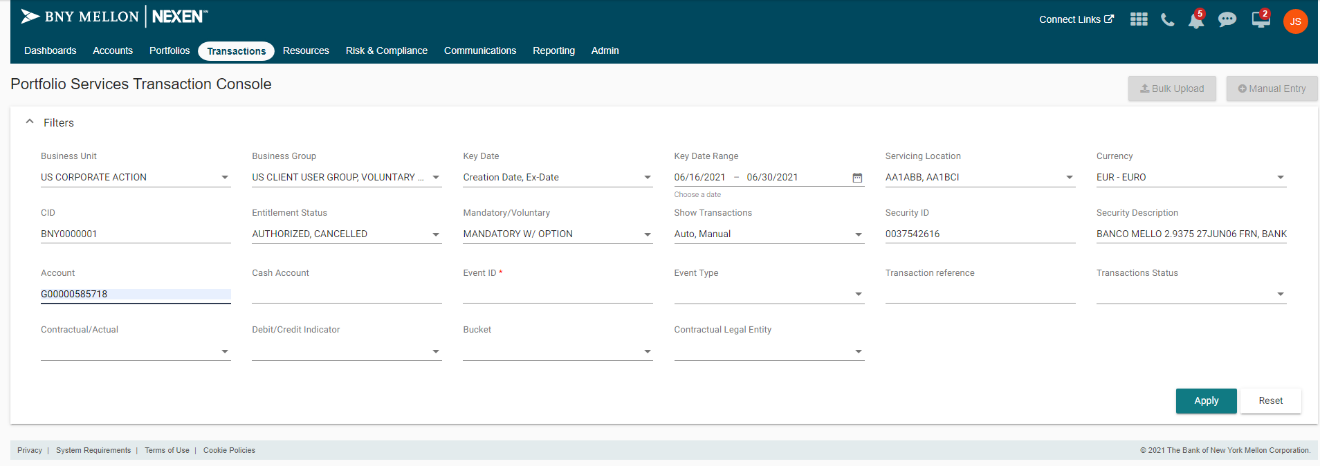
**12. Security Description** Custom Input Field

It allows user to type desired values, then fetch and show relevant records and allow user to make selection from those records.



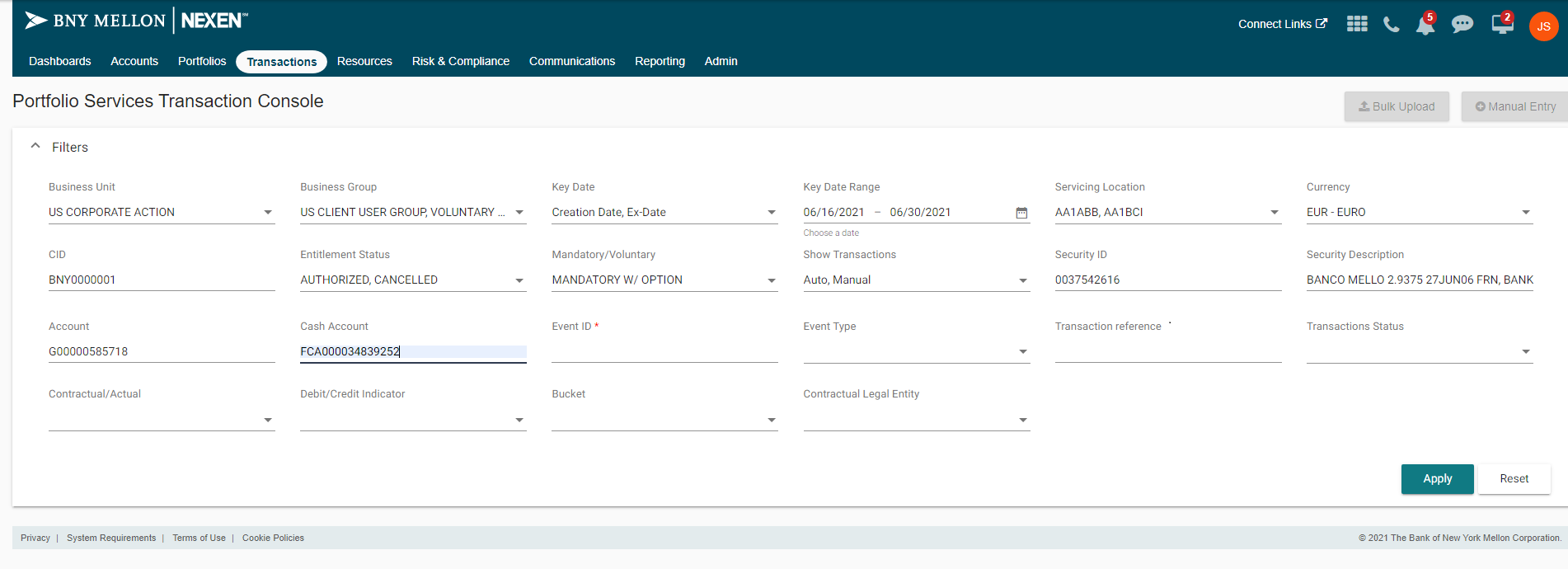
**13. Account** Input Field

It is a Input field which allows user to enter only Alpha Numeric values for this field.



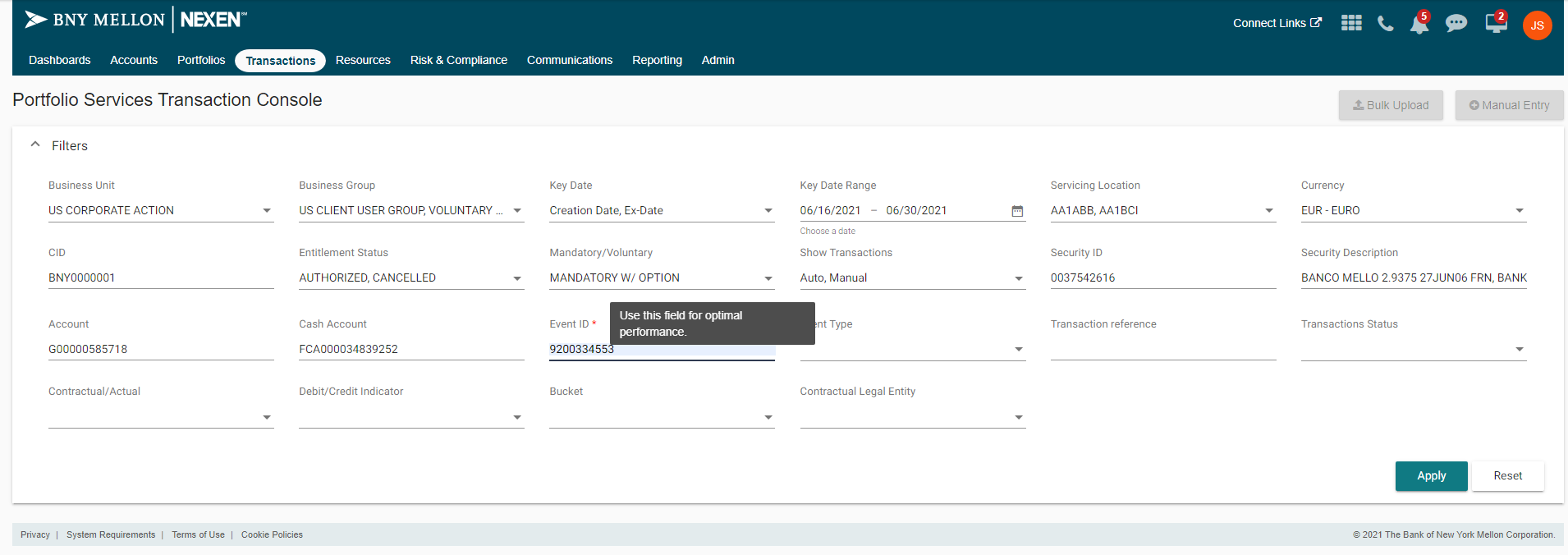
**14. Cash Account** Input Field

It is a Input field which allows user to enter only Alpha Numeric values for this field.



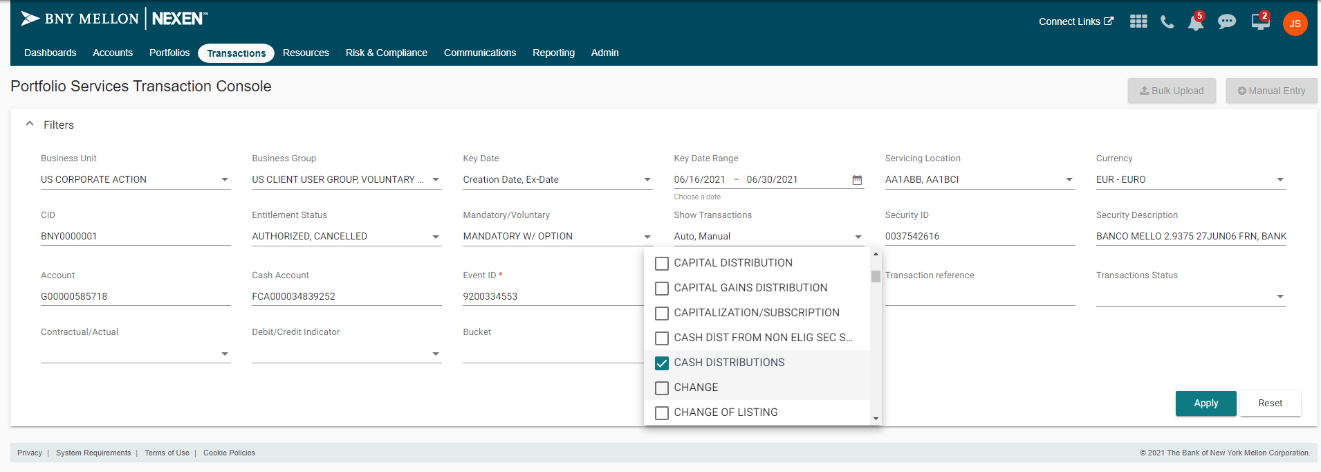
**15. Event ID** Input Field

It is a Input field which allows user to enter only Numeric values for this field.



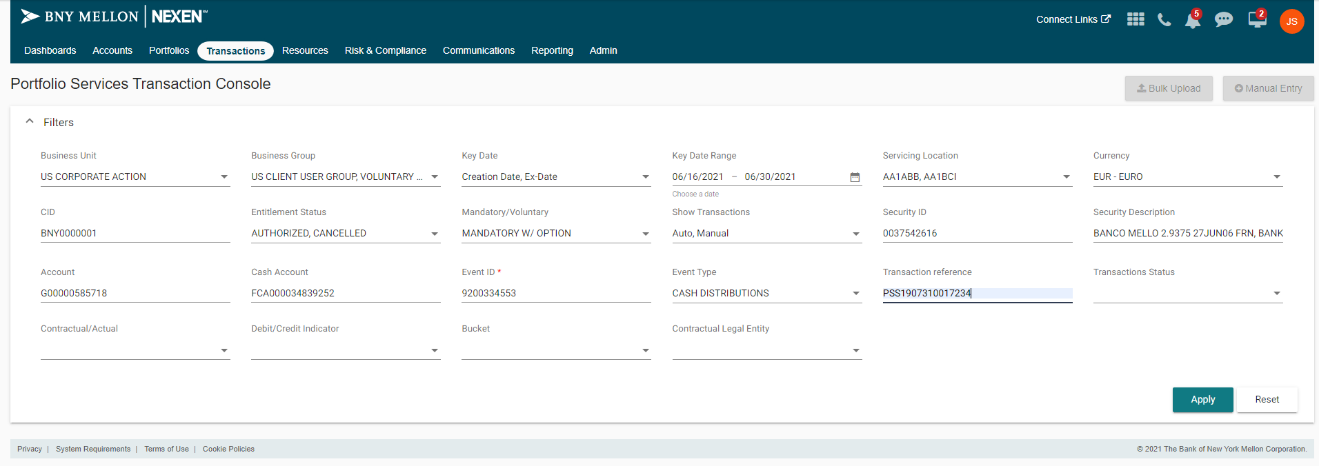
**16. Event Type** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list



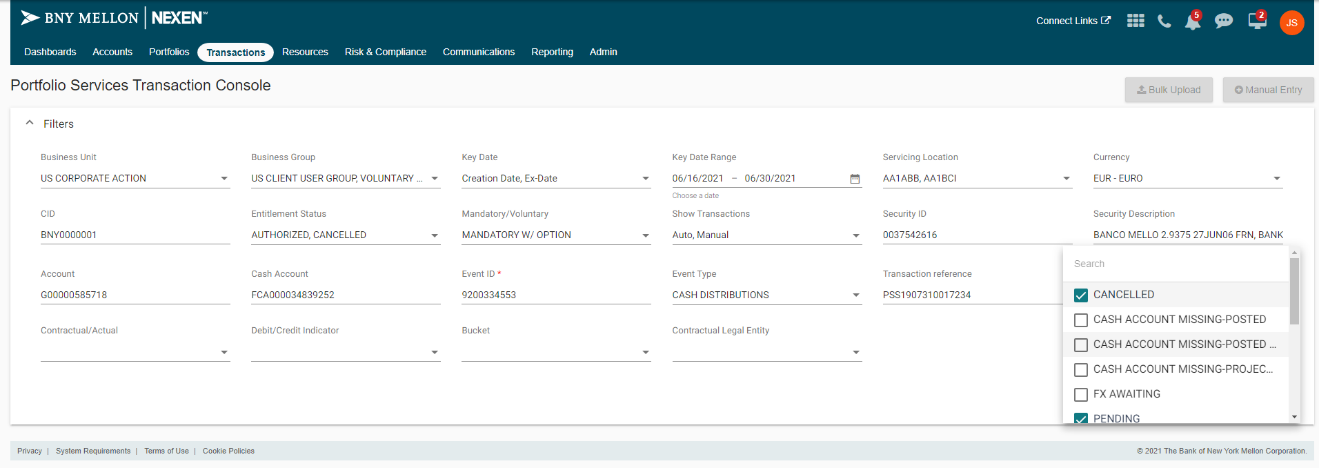
**17. Transaction reference** Input Field

It is a Input field which allows user to enter only Alpha Numeric values for this field.



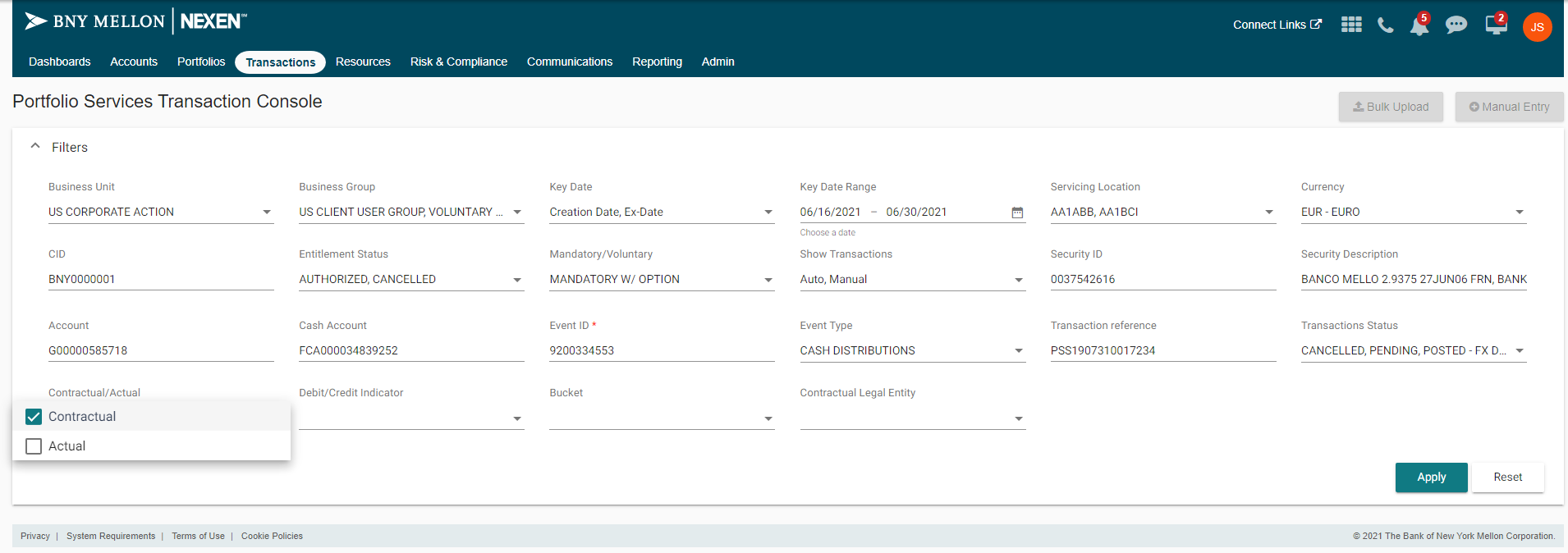
**18. Transaction status** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list



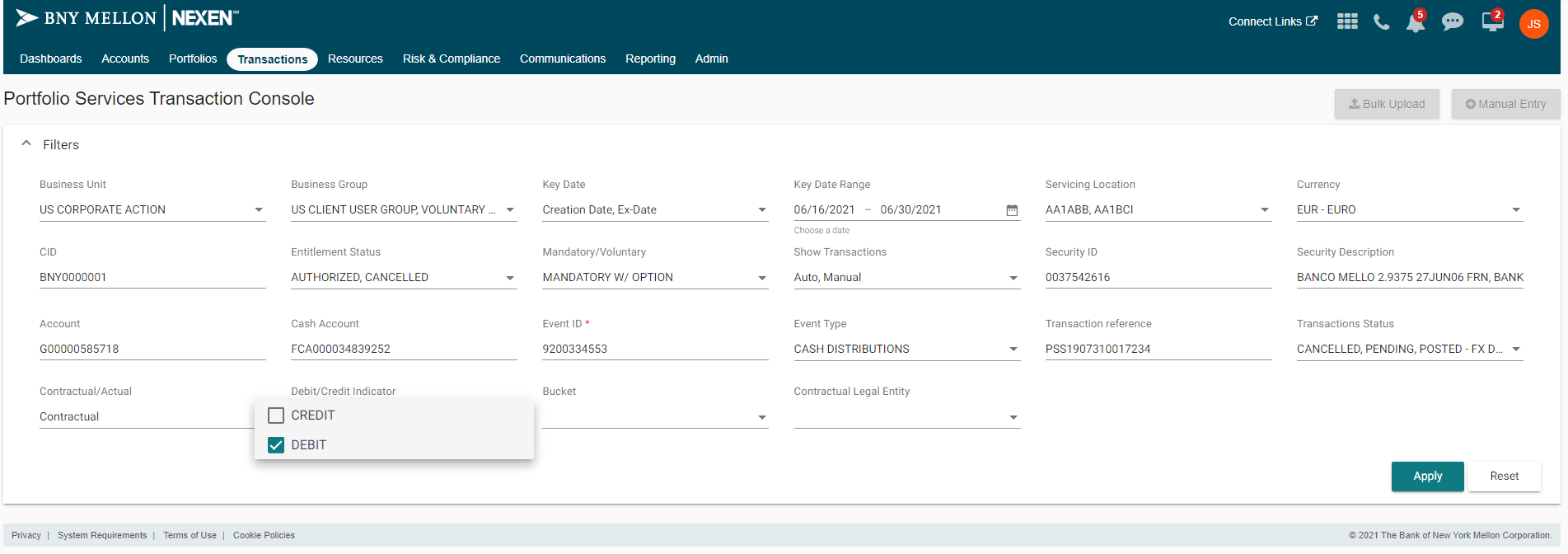
**19. Contractual/Actual** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list



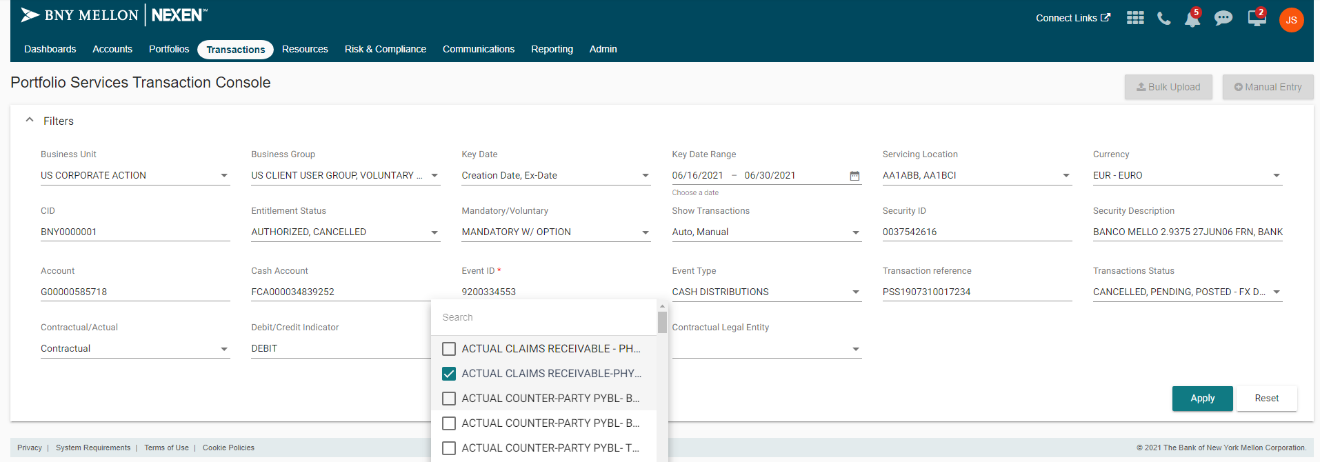
**20. Debit/Credit Indicator** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list



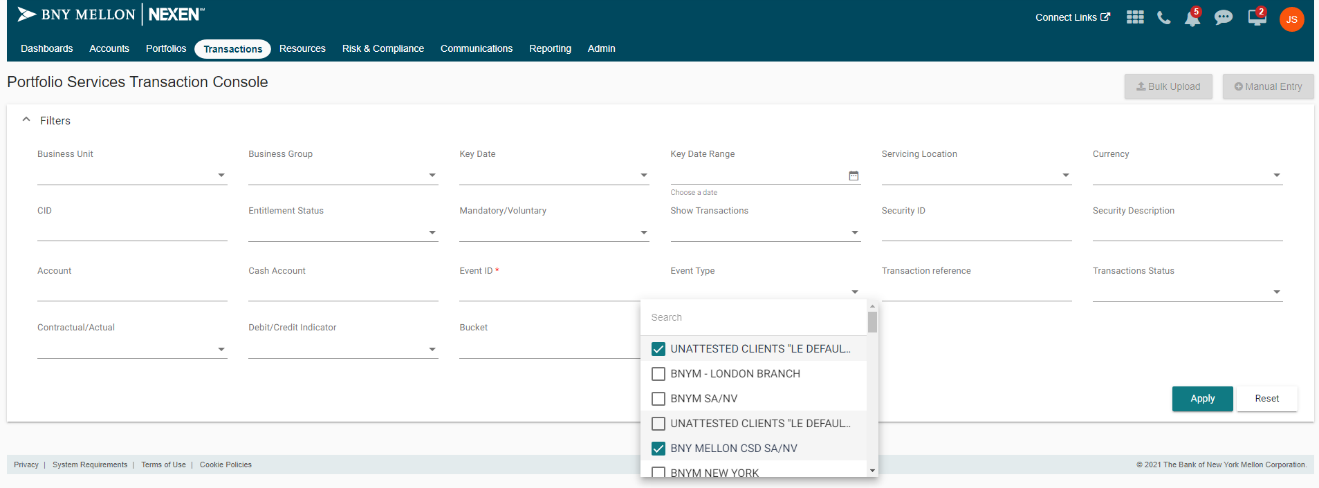
**21. Bucket** dropdown selection

It is multi valued dropdown, user can select the multiple values from dropdown list



**22. Contractual Legal Entity** dropdown selection

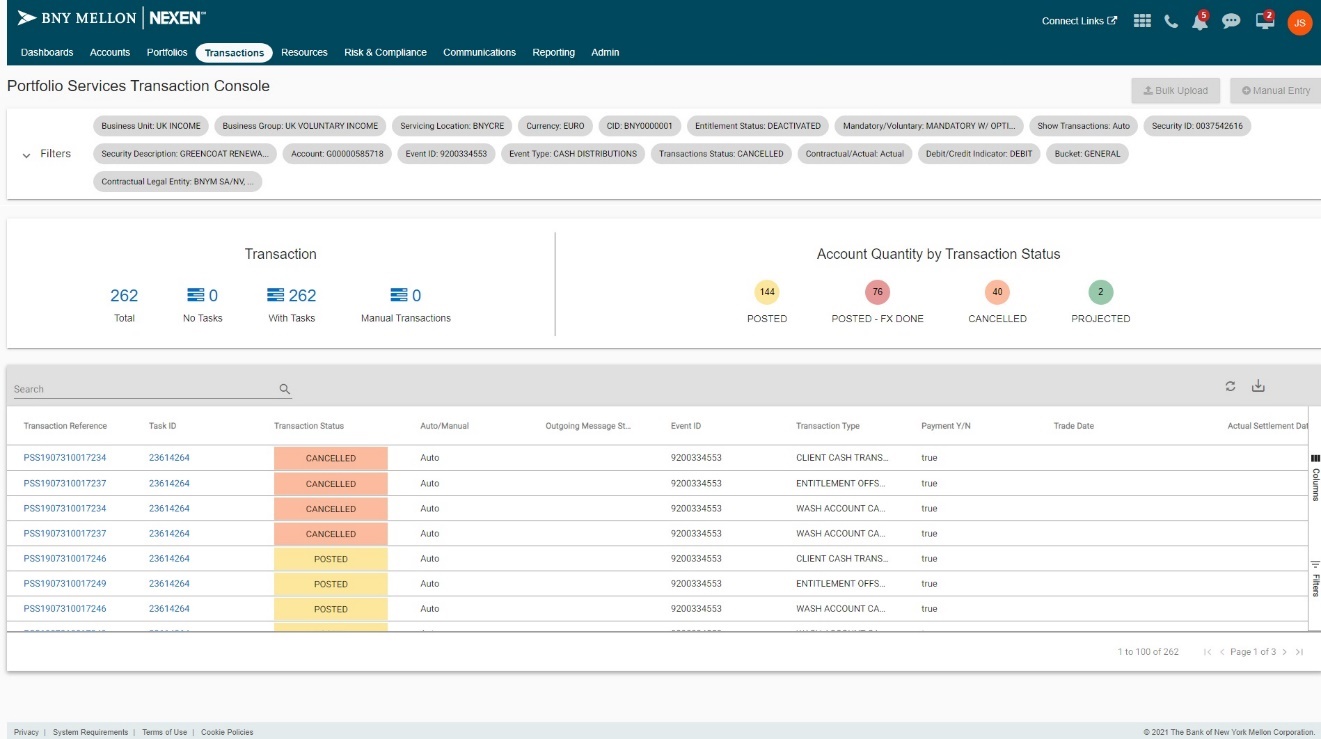
It is multi valued dropdown, user can select the multiple values from dropdown list



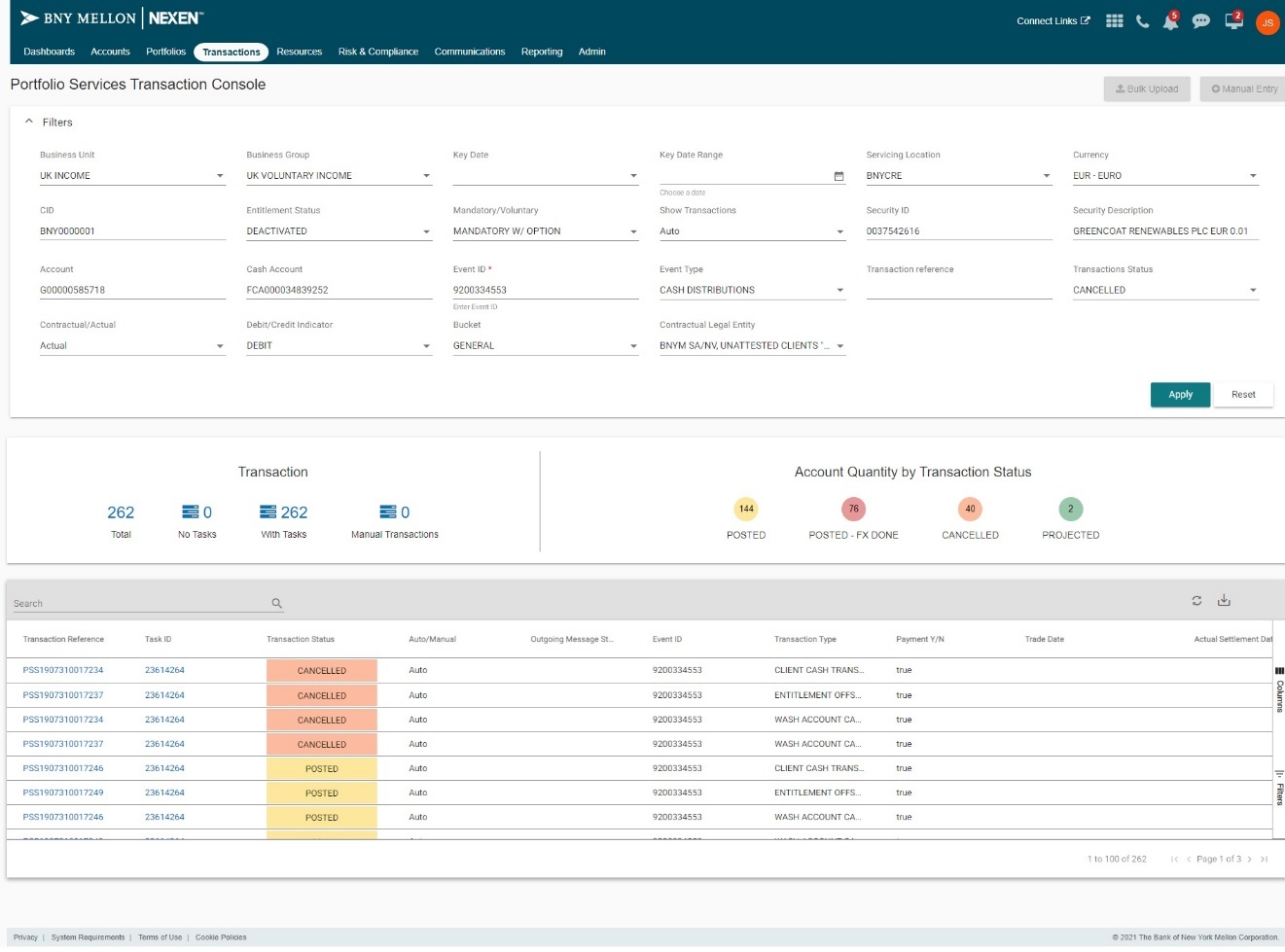
**Results**

On Applying the filters, the filter panel collapses to give room to the results.

The result is displayed in two component comprising of the Result Statistics Component and the AG-Grid Component.



User can expand the filter panel whenever they wish to do so.

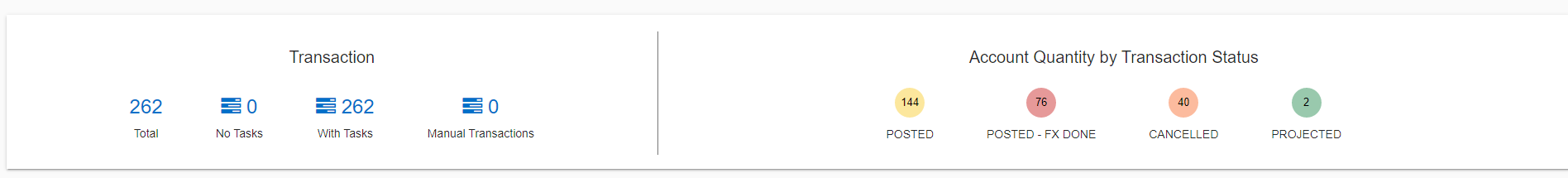


#### Transaction Statistics

Statistics Component is divided into two segments:

1] Transactions: It displays the count of total Transactions which are shown in Transaction Grid Component.   
Also displays the count of Transactions with Tasks and of those which are to be Authorized.  
2] Account Quantity by Transaction Status: It displays the count of top 5 Status Categories which are mentioned as: Projected, Cancelled, Reversed, Posted-Fx pending, Posted.

In addition to this, all these Statistics count values are clickable and perform filtering in Transaction Grid Component.



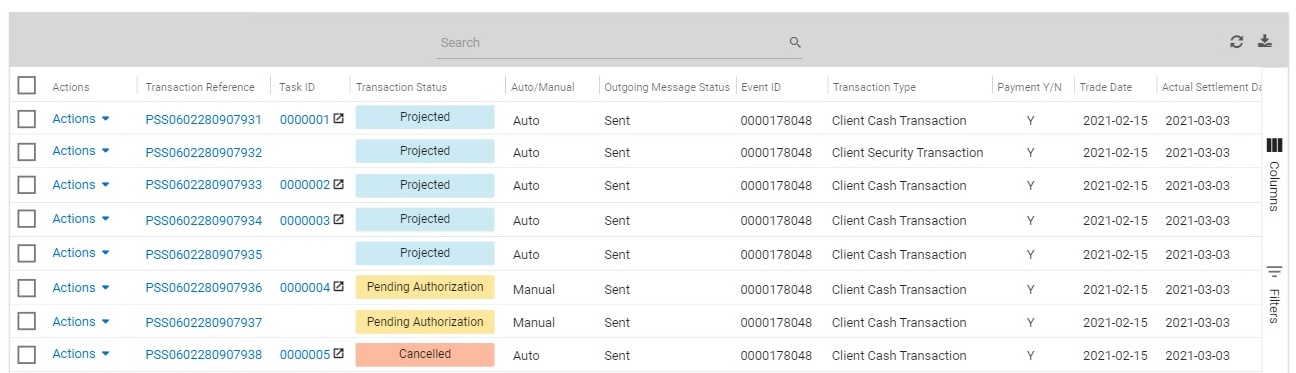
#### Transaction Grid

This Component contains all the Transactions based on the applied Filters. Also, it has an Action Dropdown per Transaction and perform actions accordingly.

There are mainly 28 column fields including Action Dropdown, out of which two fields are clickable and perform routing.

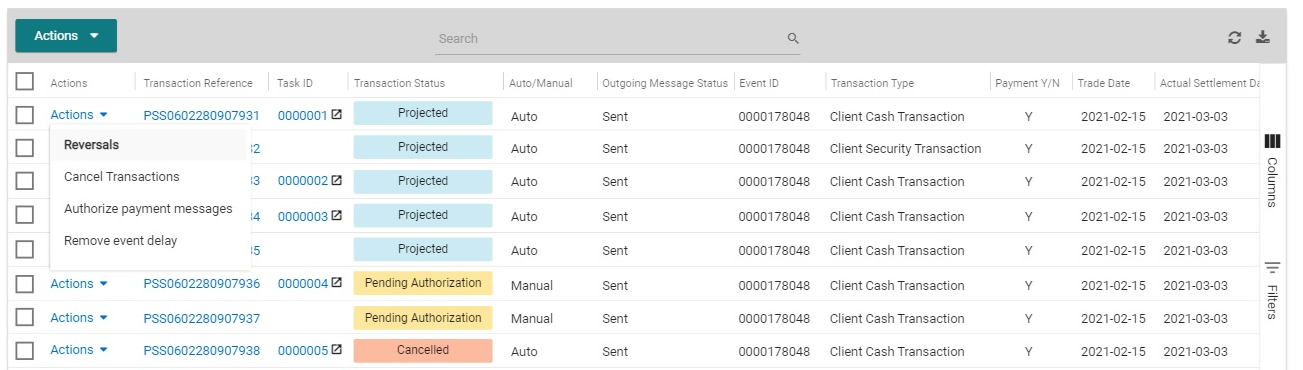
1] Transaction Reference: It takes the user to the Transaction Details page along with the Transaction Reference and Status.

2] Task ID: It takes the user to tasks plugin.



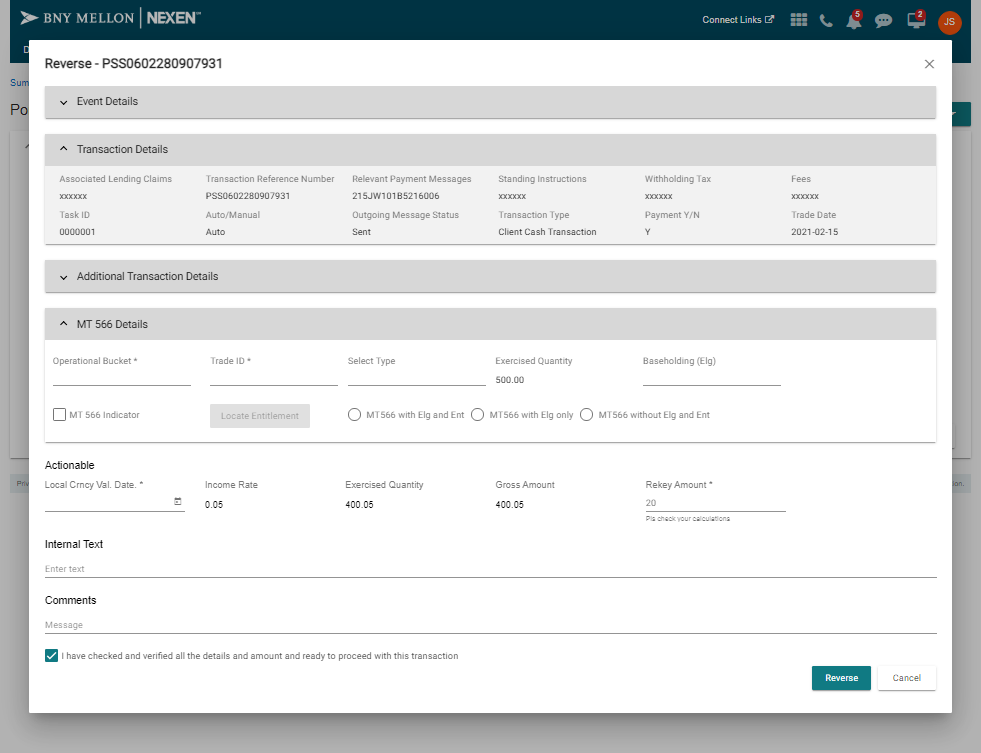
##### Action for transaction

There are mainly 4 Actions in this screen and they are part of dropdown button which is named as Actions. These Actions types are opened as dialog on the Transaction   
Details screen.



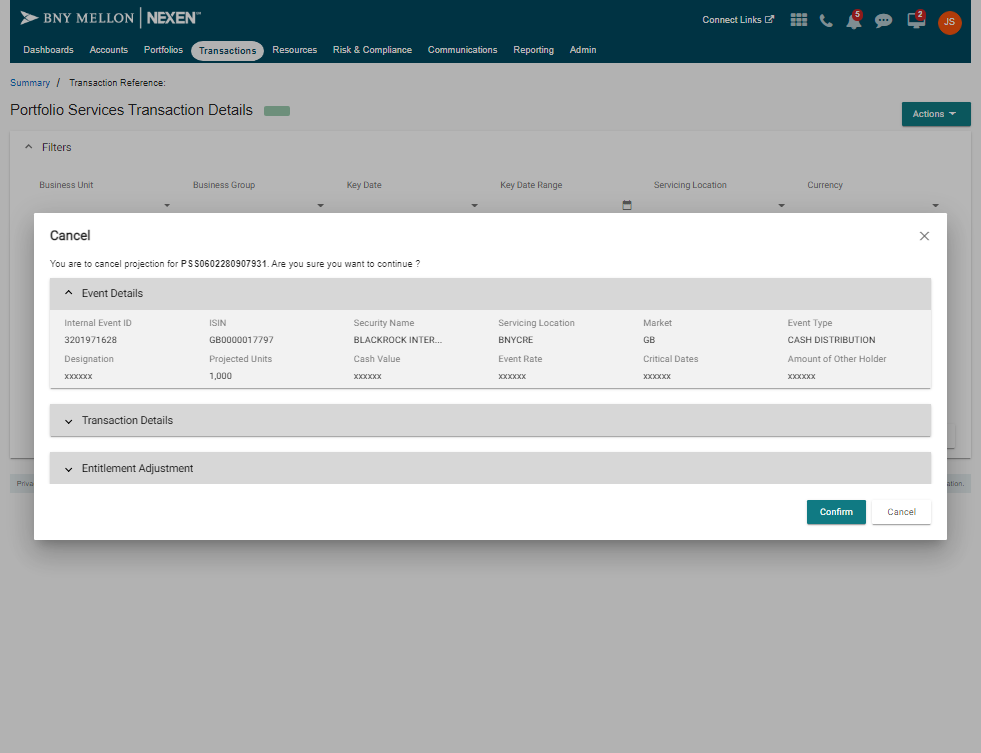
* Reverse Transaction

This dialog comprises of mainly 3 non-editable sections, 1 actionable section and 2 input fields for comments/Internal Text.  
The three non-editable sections are : Event Details, Transacytion Details, and Additional Details.  
The actionable section which is named as MT566 Details, consists of:  
a] Local Currency Value Date - An Input field  
b] Income Rate- Non-Editable field  
c] Exercised Quantity - Non-Editable field  
d] Gross Amount - Non-editable field  
e] Rekey Amount - An Input Field which will be equal to Gross amount otherwise user won't be able to submit.



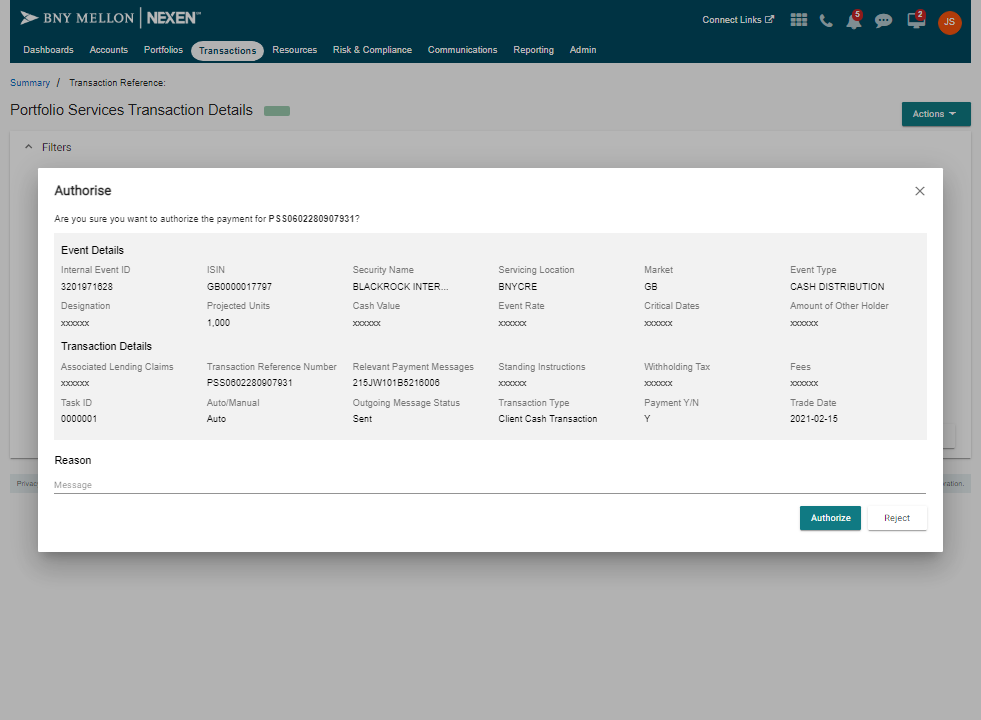
* Cancel Transaction

This dialog comprises of mainly 3 non-editable sections which are: Event Details, Transaction Details, and Entitlement Adjustment.



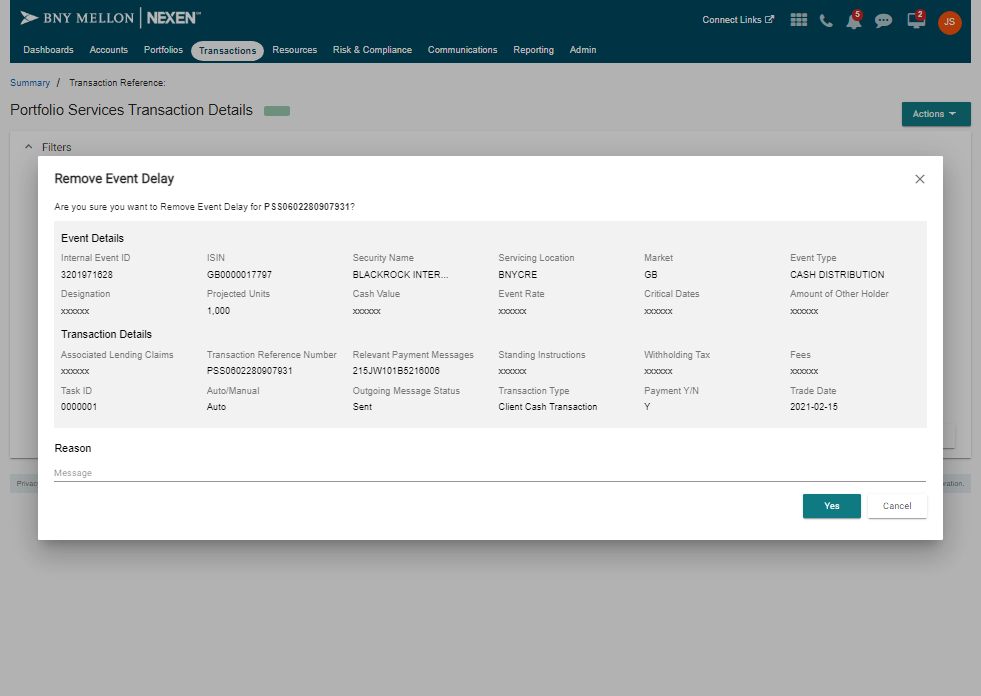
* Authorize Transaction

This dialog comprises of mainly 1 non-editable section and 1 input field for Reason/Message.



* Remove event delay

This dialog comprises of mainly 1 non-editable section.



### Transaction Details Screen

#### Event Details

#### Transaction Details

#### Related Transaction Grid

#### Audit Trails

#### Action for Transaction

### Manual Transaction

## Data Details

No Changes, Follows the current design

### Physical Database Design

No Changes, Follows the current design

### Archive and Purge consideration

No Changes, Follows the current design

### Backup and recovery strategy

No Changes, Follows the current design

## Hardware Details

No new hardware components

## Middleware Details



## Detailed Design

### Release Transaction Console – PSS

#### High Level Solution Approach

* The transactions recorded in the PSS are stored in the database. Information regarding the transaction such as (Transaction Status, Entitlement Status, Business Units/Groups, Security Id, Security Description, Debit/Credit etc...) were retrieved and populated in the screen.
* Filters in the transaction console are shown with the filters.
* Based on the selection of the filters the records are retrieved and shown in the screen.
* By applying the **Apply** button the user selected filters are send as a input to the backend and the conditions are applied and the values are fetched and populated in the screen.

|  |  |  |
| --- | --- | --- |
| PSS |  |  |
| **Reference  Requirement ID** | **Design Solution** | **Dependency/Questions** |
| Task Actions  **business-units** | Filter:   1. The details of the business units are fetched from the database using the API and shown in the screen. 2. It is a multi-valued dropdown.   API:   1. The details of the business units are fetched from the database using the API |  |
| Task Actions  **business-groups** | Filter:   1. The details of the business groups are fetched from the database using the API and shown in the screen. 2. It can be either individually or with the business-units. 3. It is a multi-valued dropdown.   API:  The details of the business groups are fetched from the database using the API |  |
| Task Actions  **mandatory-voluntary-type** | Filter:   1. This filter takes the values as a Mandatory/Voluntary. It is a multi-valued dropdown.   API:  Using the API Mandatory/Voluntary values are fetched. |  |
| Task Actions  **currency** | Filter:   1. The filter currency deals with different types of currencies that followed with different countries. 2. It is a multi-valued dropdown.   API:  Using the API the currencies values are fetched in response |  |
| Task Actions  **servicing-locations** | Filter:   1. Servicing locations filter deals with the locations. It deals with the list of location Id.   API:  Servicing locations API returns the corresponding data. |  |
| Task Actions  **transaction-status** | Filter:   1. The recorded transaction has the status to be considered as a result of each transactions.   API:  Transaction Status API returns corresponding data. |  |
| Task Actions  **entitlement-status** | Filter:   1. This filter deals with the list of entitlements allocated for the users who access the application based on the priorities.   API:  The entitlement API returns the corresponding data. |  |
| Task Actions  **debit-credit-indicator** | Filter:   1. The debit/credit are recorded for the transaction so the debit /credit it indicated in the filter column. |  |

#### Detailed Design solution Approach

* **Business Unit:**

Check if Business Group is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Business Group and Event Id are passed, then data is retrieved.

**For EX:** select txd\_bsns\_unit AS businessunit, txd\_ca\_id as eventId from prodppss.txd\_txn\_dtls where txd\_ca\_id=9200334553 and txd\_bsns\_unit='5'.

* **Business Group:**

Check if Business Unit is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Business Unit and Event Id are passed, then data is retrieved.

**For EX:** select txd\_bsns\_grp AS businessgroup, txd\_ca\_id as eventId from prodppss.txd\_txn\_dtls where txd\_ca\_id=9200334553 and txd\_bsns\_unit='304'.

* **Key date:**

Check if Key date is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Key date and Event Id are passed, then data is retrieved.

**For Ex:** SELECT cas\_effctv\_date AS keydate, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls, prodppss.cas\_ca\_sub\_evt, prodppss.cat\_corp\_actn\_tasks Where txd\_part\_id = cas\_part\_id AND txd\_ca\_id = cas\_ca\_id AND txd\_ca\_id='9200334553' AND cas\_effctv\_date='2002-07-01'.

* **Key Range :**

Check if Key date Range filters are not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Key date Range filters and Event Id are passed, then data is retrieved.

* **Service Location:**

Check if Service Location is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Service Location and Event Id are passed, then data is retrieved.

**For Ex:** select txd\_srvc\_loc AS service location, txd\_ca\_id as eventId from prodppss.txd\_txn\_dtls where txd\_ca\_id=9200334553 and txd\_srvc\_loc='BNYCRE'.

* **Currency :**

Check if Currency is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Currency and Event Id are passed, then data is retrieved.

**For Ex:** SELECT TXD\_TXN\_AMT\_CRNC AS Currency, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls

Where txd\_ca\_id='9200334553' AND TXD\_TXN\_AMT\_CRNC='EUR'

* **CID:**

Check if CID is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both CID and Event Id are passed, then data is retrieved.

**For Ex:** SELECT TXD\_CUST\_ID AS Currency, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls

Where txd\_ca\_id='9200334553' AND TXD\_CUST\_ID='BNY0000001'.

* **Entitlement Status**

Check if Entitlement Status is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Entitlement Status and Event Id are passed, then data is retrieved.

**For Ex:** SELECT TEN\_PAY\_STA AS Entitlementstatus, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls, prodppss.TEN\_TRADE\_ENTTLMNT Where TEN\_PART\_ID IN ( 1, 2, 3 ) AND TXD\_CA\_ID = TEN\_CA\_ID AND TXD\_SCA\_ID=TEN\_SCA\_ID AND TXD\_SRVC\_LOC=TEN\_SRVC\_LOC

AND txd\_ca\_id='9200334553' AND TEN\_PAY\_STA in ('1','2');

* **Mandatory/Voluntary**

Check if Mandatory/Voluntary is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Mandatory/Voluntary and Event Id are passed, then data is retrieved.

**For Ex:** SELECT CAS\_MAN\_VOL\_TYP AS MandatoryVoluntary, txd\_ca\_id AS eventid FROM

prodppss.txd\_txn\_dtls, prodppss.CAS\_CA\_SUB\_EVT Where txd\_ca\_id='9200334553' AND CAS\_MAN\_VOL\_TYP ='131' AND txd\_part\_id = cas\_part\_id AND txd\_ca\_id = cas\_ca\_id.

* **Show Transactions**

Check if Show Transactions is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Show Transactions and Event Id are passed, then data is retrieved.

**For Ex:** SELECT txd\_auto\_ind AS ShowTransactions, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls Where txd\_ca\_id='9200334553' AND txd\_auto\_ind in ('0','1').

* **Security ID**

Check if Security ID is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Security ID and Event Id are passed, then data is retrieved.

**For Ex:** SELECT txd\_sec\_id AS SecurityId, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls

Where txd\_ca\_id='9200334553' AND txd\_sec\_id ='0037542616'

* **Security Description**

Check if Security Description is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Security Description and Event Id are passed, then data is retrieved.

**For Ex:** SELECT SEC\_SHRT\_NAME AS SecurityDescription, txd\_ca\_id AS eventid

FROM prodppss.txd\_txn\_dtls, prodppss.SEC\_SEC Where txd\_ca\_id='9200334553'

AND SEC\_SHRT\_NAME = 'GREENCOAT RENEWABLES PLC EUR 0.01' AND TXD\_SEC\_ID = SEC\_SEC\_ID.

* **Account**

Check if Account is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Account and Event Id are passed, then data is retrieved.

**For Ex:** SELECT txd\_sca\_id AS Account, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls Where

txd\_ca\_id='9200334553' AND txd\_sca\_id = 'G00000585718'

* **Cash Account**

Check if Cash Account is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Cash Account and Event Id are passed, then data is retrieved.

**For Ex:** SELECT TXD\_CSH\_ACCT\_ID AS CashAccountId, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls Where txd\_ca\_id='9200334553' AND TXD\_CSH\_ACCT\_ID = 'SPNS00000000237'

* **Event ID**

Check if Event ID is not null.

If Event Id is passed, results are fetched based on the particular Event Id.

If both Currency and Event Id are passed, then data is retrieved.

**For Ex:** select txd\_ca\_id as eventId from prodppss.txd\_txn\_dtls where txd\_ca\_id=9200334553

* **Event Type**

Check if Event Type is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Event Type and Event Id are passed, then data is retrieved.

**For Ex:** SELECT txd\_ca\_typ AS EventType, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls

Where txd\_ca\_id='9200334553' AND txd\_ca\_typ ='11'.

* **Transaction Reference**

Check if Transaction Reference is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Transaction Reference and Event Id are passed, then data is retrieved.

**For Ex:** SELECT TXD\_DESTN\_REF\_NUM AS TransactionReference, txd\_ca\_id AS eventid

FROM prodppss.txd\_txn\_dtls Where txd\_ca\_id='9200334553'AND TXD\_DESTN\_REF\_NUM ='PSS1907310017237'.

* **Transaction Status**

Check if Transaction Status is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Transaction Status and Event Id are passed, then data is retrieved.

**For Ex:** SELECT txd\_txn\_sta AS TransactionStatus, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls

Where txd\_ca\_id='9200334553' AND txd\_txn\_sta ='5'.

* **Contractual/Actual**

Check if Contractual/Actual is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Contractual/Actual and Event Id are passed, then data is retrieved.

**For Ex:** SELECT TXD\_CNTRA\_IND AS ContractualActual, txd\_ca\_id AS eventid FROM

prodppss.txd\_txn\_dtls Where txd\_ca\_id='9200334553' AND TXD\_CNTRA\_IND in ('1','2').

* **Credit/Debit Indicator**

Check if Contractual/Actual is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Contractual/Actual and Event Id are passed, then data is retrieved.

**For Ex:** SELECT TXD\_DR\_CR\_IND AS CreditDebitIndicator, txd\_ca\_id AS eventid FROM prodppss.txd\_txn\_dtls Where txd\_ca\_id='9200334553' AND TXD\_DR\_CR\_IND in ('1','2').

* **Bucket**

Check if Bucket is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Bucket and Event Id are passed, then data is retrieved.

**For Ex:** SELECT TXD\_OPER\_BCKT AS bucket, txd\_ca\_id AS eventid FROM

prodppss.txd\_txn\_dtls Where txd\_ca\_id='9200334553' AND TXD\_OPER\_BCKT = '1'.

* **Contractual Legal Entity**

Check if Contractual Legal Entity is not null.

If Event Id alone is passed, results are fetched based on Event Id alone.

If both Contractual Legal Entity and Event Id are passed, then data is retrieved.

**API Part:**

* **Mandatory-voluntary-type**

URL: /txn/meta/mandatory-voluntary-type/list

Data population:

Query: SELECT SCO\_CODE\_VALUE as id, SCO\_LONG\_DESC as description FROM PRODPPSS.SCO\_SYSTEM\_CODES WHERE SCO\_CODE\_TYPE = 1763 ORDER BY description.

* **Currency**

URL: /txn/meta/currency/list

Data population:

Query: SELECT SCO\_CODE\_VALUE as id, SCO\_LONG\_DESC as description FROM PRODPPSS.SCO\_SYSTEM\_CODES WHERE SCO\_CODE\_TYPE = 34 ORDER BY description.

* **Servicing-locations**

URL: txn/meta-info/servicing-locations/list

Data is retrieved.

Query: SELECT DISTINCT SUB\_SRVC\_LOC FROM PRODPPSS.SUB\_ACCT\_DTLS ORDER BY SUB\_SRVC\_LOC.

* **Business-unit**

URL: /txn/meta/business-unit/list

Data is retrieved.

Query: SELECT SCO\_CODE\_VALUE as id, SCO\_LONG\_DESC as description FROM PRODPPSS.SCO\_SYSTEM\_CODES WHERE SCO\_CODE\_TYPE = 1411 ORDER BY description

* **Business-group**

URL: /txn/meta/business-group/list

Data is retrieved.

Query: SELECT SCO\_CODE\_VALUE as id, SCO\_LONG\_DESC as description FROM PRODPPSS.SCO\_SYSTEM\_CODES WHERE SCO\_CODE\_TYPE = 1705 ORDER BY description.

* **Transaction-status**

URL: /txn/meta/transaction-status/list

Data is retrieved.

Query: SELECT SCO\_CODE\_VALUE as id, SCO\_LONG\_DESC as description FROM PRODPPSS.SCO\_SYSTEM\_CODES WHERE SCO\_CODE\_TYPE = 1473 ORDER BY description.

* **Entitlement-status**

URL: /txn/meta/pay-status/list

Data is retrieved.

Query: SELECT SCO\_CODE\_VALUE as id, SCO\_LONG\_DESC as description FROM PRODPPSS.SCO\_SYSTEM\_CODES WHERE SCO\_CODE\_TYPE = 366 ORDER BY description.

* **Event-type**

URL:/txn/meta/event-type/list

Data is retrieved.

Query: SELECT SCO\_CODE\_VALUE as id, SCO\_LONG\_DESC as description FROM PRODPPSS.SCO\_SYSTEM\_CODES WHERE SCO\_CODE\_TYPE = 1763 ORDER BY description

* **Bucket**

URL:/txn/meta/bucket/list

Data is retrieved.

Query: SELECT SCO\_CODE\_VALUE as id, SCO\_LONG\_DESC as description FROM PRODPPSS.SCO\_SYSTEM\_CODES WHERE SCO\_CODE\_TYPE = 1406 ORDER BY description

Debit-credit-indicator:

URL: /txn/meta/debit-credit-indicator/list

Data is retrieved.

Query: SELECT SCO\_CODE\_VALUE as id, SCO\_LONG\_DESC as description FROM PRODPPSS.SCO\_SYSTEM\_CODES WHERE SCO\_CODE\_TYPE = 22 ORDER BY description

* **Contractual Legal Entity**

URL: /txn/meta-info/contractual-legal-entity/list

Data is retrieved.

Query: SELECT distinct t2.CTRCT\_LEI as id,TRIM(t3.BPT\_FULL\_NAME) as description FROM PRODPPSS.ACT\_ACT\_MSTR t1,PRODPPSS.A\_TWG\_ACCT\_DTL t2,PRODPPSS.BPT\_BP t3,PRODPPSS.BPT\_BP t4 WHERE REPLACE(t1.ACT\_SCA\_ID, 'G', '00') = t2.SRC\_APPL\_ACCT\_NO AND t3.BPT\_BP\_ID = 'LE'||COALESCE(t2.CMP\_ID,'99999') AND t4.BPT\_BP\_ID = 'LE'||COALESCE(t2.CTRCT\_LEI,'99999') ORDER BY t2.CTRCT\_LEI DESC.

* **Security Descriptions**

URL: /txn/meta-info/security-descriptions/lookup?contains=america

Data is Retrieved.

Query: SELECT DISTINCT(SEC\_SHRT\_NAME) FROM PRODPPSS.SEC\_SEC WHERE UPPER(SEC\_SHRT\_NAME) LIKE '%AMERICA%' ORDER BY SEC\_SHRT\_NAME FETCH FIRST 200 ROWS ONLY.

### Transaction Console mapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.No** | **Transaction Console Parameters** | **Column Name** | **Table Name** | **Field Parameter Name** |
| 1 | Transaction Reference | TXD\_DESTN\_REF\_NUM | TXD\_TXN\_DTLS | transactionId |
| 2 | Task Id | CAT\_TASK\_ID | CAT\_CORP\_ACTN\_TASKS | taskID |
| 3 | Transaction Status | TXD\_TXN\_STA | TXD\_TXN\_DTLS | transactionStatusId |
| 4 | Outgoing Message Status | TXD\_OUT\_GNG\_STA | TXD\_TXN\_DTLS | outgoingMessageStatusId |
| 5 | Event ID | TXD\_CA\_ID | TXD\_TXN\_DTLS | eventId |
| 6 | Transaction Type | TXD\_TXN\_CATEGORY | TXD\_TXN\_DTLS | categoryId |
| 7 | Payment Y/N | Payment Y/N: {(TXD\_TXN\_STA in (2,10,11,12)) AND (TXD\_ERM\_REF\_NUM IS NOT NULL) AND (TXD\_CNTRA\_IND = 2)} | TXD\_TXN\_DTLS | paymentIndicator |
| 8 | Trade Date | TXD\_TRADE\_DT | TXD\_TXN\_DTLS | tradeDate |
| 9 | Actual Settlement Date | TXD\_ACT\_STLMNT\_DT | TXD\_TXN\_DTLS | actualSettlementDate |
| 10 | Contracting Legal Entity | BPT\_FULL\_NAME | BPT\_BP | contractingLegalEntityDescription |
| 11 | Value Date | TXD\_LCL\_CRN\_VAL\_DT | TXD\_TXN\_DTLS | localCurrencyValueDate |
| 12 | Pay Date | TXD\_EVNT\_PAY\_DT | TXD\_TXN\_DTLS | payDate |
| 13 | ERD | CAS\_EXP\_TO\_RCV\_DT | CAS\_CA\_SUB\_EVT | erd |
| 14 | Event Type | TXD\_CA\_TYP | TXD\_TXN\_DTLS | eventType |
| 15 | ISIN | CAS\_SEC\_ID\_ISIN | CAS\_CA\_SUB\_EVT | isin |
| 16 | Security Name | TOD\_SEC\_NAME | TOD\_TXN\_OTHR\_DTLS | securityName |
| 17 | Account Number | TXD\_SCA\_ID | TXD\_TXN\_DTLS | clientAccountId |
| 18 | Units | TXD\_BASE\_HLDNG | TXD\_TXN\_DTLS | baseHolding |
| 19 | Transaction Amount | TXD\_TXN\_AMOUNT | TXD\_TXN\_DTLS | amount |
| 20 | Currency | TXD\_TXN\_AMT\_CRNC | TXD\_TXN\_DTLS | sourceCurrency |
| 21 | SVCLOC | TXD\_SRVC\_LOC | TXD\_TXN\_DTLS | servicingLocation |
| 22 | BUCKET | TXD\_OPER\_BCKT | TXD\_TXN\_DTLS | operationalBucketId |
| 23 | CSD | CAS\_EXP\_TO\_RCV\_DT | CAS\_CA\_SUB\_EVT | csd |
| 24 | Standing Instruction Y/N | ‘TXD\_PYMNT\_MTD’ field is populated then SI is Y else N | TXD\_TXN\_DTLS | standingInstructionIndicator |
| 25 | Risk Score |  |  | riskScore |
| 26 | AutoManalIndicator | TXD\_AUTO\_IND | TXD\_TXN\_DTLS | manualAutoIndicator |

## Software Details

### Frontend Software

* Angular - 12.0.3
* Node JS - 12.22.1
* Node Package Manager (npm) - 6.14.12
* Grunt- v1.4.3

### Backend Software

* lombok:1.18.16
* Java 11
* Intellij (Ultimate)
* Springboot version:2.1.7
* maven-resources-plugin:3.1.0
* Appdynamics (building the docker image)
* GitLab Enterprise Edition 13.11.5

## Error Handling

**Introduction :**

* The NEXEN UI Platform allows and leverages the Karma/Jasmine framework for code unit testing. The platform also provides a Coverage Summary/Report based on how the units of code were tested.
* Angular spec files are TypeScript files and reside in the same folder as components and services (/WebContent/modules/<componentName>). The Karma Test Runner is configured to get all the Angular spec files from either the core or the plugin's /WebContent/modules directory.

**Unit Testing an Angular Service:**

* **Transaction-console.service.ts**

import { Injectable } from "@angular/core";

import { PssHttpClient } from "../service/pss-http-client.service";

import { Subject } from "rxjs";

**// For EX:-**

@Injectable({

providedIn: "root",

})

export class TransactionConsoleService{

constructor(private transactionHttpClient: PssHttpClient) {

// Placeholder

}

loadBusinessUnits(): Subject<any> {

const subject = new Subject<any>();

this.transactionHttpClient.getBusinessUnits()

.subscribe(

(response: any) => this.handleBusinessUnitsResponse(response, subject),

(error: any) => subject.error(error)

);

return subject;

}

}

* **Transaction-console.service.spec.ts**

import { TestBed } from "@angular/core/testing";

import { PssHttpClient } from "../service/pss-http-client.service";

import { TransactionConsoleService } from "./transaction-console.service";

import { of, Subject, throwError } from "rxjs";

**//For EX:-** Angular Unit Test for Transaction-console.service Service

describe("Transaction Console Service", () => {

let service: any;

beforeEach(() => {

TestBed.configureTestingModule({

providers: [

TransactionConsoleService,

{

provide: PssHttpClient, useValue:{

getBusinessUnits: () => of({})

}

}

]

});

});

beforeEach(() => {

service = TestBed.get(TransactionConsoleService); // Provides an injectable for the service.

});

describe("loadBusinessUnits", () => {

it("should properly load the Business Unit list via API", () => {

spyOn(service, "handleBusinessUnitsResponse").and.returnValue(null);

service.loadBusinessUnits();

expect(service.handleBusinessUnitsResponse).toHaveBeenCalledTimes(1);

spyOn(service["transactionHttpClient"], "getBusinessUnits").and.returnValue(throwError({}));

service.loadBusinessUnits();

expect(service.handleBusinessUnitsResponse).toHaveBeenCalledTimes(1);

});

});

});

**Unit Testing an Angular Component:**

* **transaction-console.component.ts**

import { Component,ViewChild } from "@angular/core";

import { FiltersComponent } from "./filters/transaction-filters.component";

import { AgGridComponent } from "../common/ag-grid/ag-grid.component";

@Component({

selector: "transaction-summary",

templateUrl: "./transaction-console.component.html",

styleUrls: ["./transaction-console.component.css"]

})

**For Ex -** Angular Component

export class TransactionConsoleComponent<D> {

@ViewChild(FiltersComponent) filterComponent: FiltersComponent<D>;

@ViewChild(AgGridComponent) agGridComponent: AgGridComponent<D>;

constructor() {}

handleInquiryError(error: any): void {

this.showError({ messageText: "Failed to load event data. Please try again later.", errorDetails: error });

this.agGridComponent.handleGridReset();

this.filterComponent.togglePanelOpen(true);

}

}

* **transaction-console.component.spec.ts**

// **Tests the Transaction Console Component**

import { NO\_ERRORS\_SCHEMA } from "@angular/core";

import { ComponentFixture, fakeAsync, TestBed } from "@angular/core/testing";

import { TransactionConsoleComponent } from "./transaction-console.component";

describe("Transaction Console Component", () => {

let fixture: ComponentFixture<TransactionConsoleComponent<any>>;

let component: TransactionConsoleComponent<any>;

beforeEach(fakeAsync(() => {

TestBed.configureTestingModule({

declarations: [

TransactionConsoleComponent

],

imports: [],

providers: [],

schemas: [NO\_ERRORS\_SCHEMA]

})

.compileComponents();

}));

beforeEach(() => {

const filterComponent = jasmine.createSpyObj("FiltersComponent", ["performQuickLookup", "togglePanelOpen", "applyFilters"]);

const gridApi = jasmine.createSpyObj("gridApi", ["setRowData", "collapseAll", "addEventListener", "removeEventListener", "setQuickFilter", "setFilterModel", "handleGridReset", "setInitialParameters", "handleGridResponse"]);

fixture = TestBed.createComponent(TransactionConsoleComponent);

component = fixture.componentInstance;

component.filterComponent = filterComponent;

component.agGridComponent = gridApi;

});

it("should properly initialize", () => {

expect(fixture).toBeDefined();

expect(component).toBeDefined();

});

describe("handleInquiryError", () => {

it("should handle errors properly", () => {

spyOn(component, "showError").and.returnValue(null);

//component.handleInquiryError({ metadata: { success: false }, data: null, error: {} });

const error: any = { };

component.handleInquiryError(error);

expect(component.showError).toHaveBeenCalled();

expect(component.agGridComponent.handleGridReset).toHaveBeenCalled();

expect(component.filterComponent.togglePanelOpen).toBeDefined();

});

});

});

**Command:**

grunt nxn-plugin:test:<plugin\_name>:angular-Runs specified plugin's Angular spec files

**Results**

Karma results are located in the test-results/ directory.

**Error Codes for Backend**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error Description** | **Codes** | **Message** |
| 1 | Successful Fetch | 200 | Success |
| 2 | The request is not successful | 400 | BadRequest |
| 3 | The request is not authorized | 401 | Unauthorized |
| 4 | The request is forbidden | 403 | Forbidden |
| 5 | The requested reesource is unavailable | 404 | Not Found |
| 6 | The internal error message | 500 | Internal Server Error |
|  |  |  |  |

## System Interface Details

NA

## Network Details

There are no network changes to be done as a part of this project.

# Code Quality Reviews

All code modules are required to undergo some level of review. This table lists the code modules that will undergo the most formal kind of review - Peer Review.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Service Names** | **Severity** | **Code Aspect** | **Code Requirement** | **Suggested Code Comments** | **Status** |
| TransactionConsole | M | Deployment | App Engine Deployment | In docker file with respect to snapshot version and naming convention(gpsimtxn-api) was suggested in docker file.The assembly file name such as name,app,orgid was changed as per the naming convention suggested(name: "gpsimtxn",family: "pss",app: "gpsimtxn",org-id: "pss").The hubble image generation was posted in the assembly.yaml file with the naming convention "(hubble.bnymellon.net:5000/bnym/linux/pss/pss/gpsimtxn)" .In Spring Boot configuration parameter the new registry was mapped in the naming convention "(cluster.bnymellon.net/dev-pss-gpsimregistry-eureka/eureka)". | Done & Deployed |
| Entity Manager-TC | M | Deployment | App Engine Deployment | In docker file with respect to snapshot version and naming convention(gpsim-entity-manager-api) was suggested in docker file.The assembly file name such as name,app,orgid was changed as per the naming convention suggested(name: "gpsimentitymanager",family: "pss",app: "gpsimentitymanager",org-id: "pss").The hubble image generation was posted in the assembly.yaml file with the naming convention "(hubble.bnymellon.net:5000/bnym/linux/pss/pss/gpsimentitymanager)" .In Spring Boot configuration parameter the new registry was mapped in the naming convention "(cluster.bnymellon.net/dev-pss-gpsimregistry-eureka/eureka)". | Done & Deployed |
| Cache-TC | M | Deployment | App Engine Deployment | In docker file with respect to snapshot version and naming convention(gpsim-distributed-cache-api) was suggested in docker file.The assembly file name such as name,app,orgid was changed as per the naming convention suggested(name: "gpsimcache",family: "pss",app: "gpsimcache",org-id: "pss").The hubble image generation was posted in the assembly.yaml file with the naming convention "(hubble.bnymellon.net:5000/bnym/linux/pss/pss/gpsimcache)" .In Spring Boot configuration parameter the new registry was mapped in the naming convention "(cluster.bnymellon.net/dev-pss-gpsimregistry-eureka/eureka)". | Done & Deployed |
| Registry-TC |  | Deployment | App Engine Deployment | In docker file with respect to snapshot version and naming convention(gpsim-registry-api) was suggested in docker file.The assembly file name such as name,app,orgid was changed as per the naming convention suggested(name: "gpsimregistry",family: "pss",app: "gpsimregistry",org-id: "pss").The hubble image generation was posted in the assembly.yaml file with the naming convention "(hubble.bnymellon.net:5000/bnym/linux/pss/pss/gpsimregistry)" .In Spring Boot configuration parameter the new registry was mapped in the naming convention "(cluster.bnymellon.net/dev-pss-gpsimregistry-eureka2/eureka)". | Done & Deployed |
| Information-TC | M | Deployment | App Engine Deployment | In docker file with respect to snapshot version and naming convention(gpsim-registry-api) was suggested in docker file.The assembly file name such as name,app,orgid was changed as per the naming convention suggested(name: "gpsimregistry",family: "pss",app: "gpsimregistry",org-id: "pss").The hubble image generation was posted in the assembly.yaml file with the naming convention "(hubble.bnymellon.net:5000/bnym/linux/pss/pss/gpsimregistry)" .In Spring Boot configuration parameter the new registry was mapped in the naming convention "(cluster.bnymellon.net/dev-pss-gpsimregistry-eureka2/eureka)". | Done & Deployed |
| Transaction Search Filter | M | Transaction Service | Code level change | Used Legecy java code changed according to java 11 features and compatabile. | Done |
| Transaction Filter | M | Transaction Service | Code level change | Java Streams were used instead of foreach loop | Done |
| Code Optimization | M | Code change | Code level change | Removed boiler plate codes in service layer for transaction console service | Done |
| Performance Optimization | M | Code change | Code level change | Performance optimized on all (DAO-related to transaction console) indexed columns in the database for faster retrieval | Done |
| Log Management | M | Code change | Code level change | Added loggers for debug as well as info logs in all layers using SLF4J. | Done |
| Naming Conventation | L | Code change | Code level change | Guided with naming convention in all the classes,methods and parameters | Done |

# Logging Policy

Using the SLF4J details of all the logging mechanism is incorporated in the code level.

This is helpful during the debugging & process in all environments including the production.

Required Annotation:@Slf4j.

# Implementation Considerations

No additional implementation consideration exists for this project.

# Contributors

|  |  |  |  |
| --- | --- | --- | --- |
| Role | | Name | Organization |
| Key Customer Contact(s) | |  |  |
| Project Manager/Supervisor (if applicable) |  |  |
| Backup Project Manager/Supervisor (if applicable) | |  |  |
| Business Analyst (indicate if author when applicable) | |  |  |
| System Analyst (indicate if author when applicable) | |  |  |
| Vendor Project Manager  (if applicable) | |  |  |
| Backup Vendor Project Manager (if applicable) | |  |  |
| Systems Architect (if applicable) | |  |  |
| Data Architect (if applicable) | |  |  |
| Network Architect (if applicable) | |  |  |
| System Area SME (if applicable) | |  |  |
| Developers  (if applicable) | |  |  |

# Terms and Abbreviations

|  |  |
| --- | --- |
| Term / Abbreviation | Meaning |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Support and Reference Documents

# Appendix A – On Line Specifications

|  |  |
| --- | --- |
| Screen Name | Description |
| Users | Define who will use the function. See also Security paragraph under Operational Requirements. |
| Function | Define the function of the screen/updated field/etc. (i.e.: Add, Maintain, Delete, Inquiry, List. Specify if calculations are needed, i.e. subtotal, total, etc. |
| Required Fields | Length (if known, i.e. alphanumeric, alpha only, etc. |
| Optional Fields | Length (if known, i.e. alphanumeric, alpha only, etc. |
| Sort Criteria | Document the sort sequence for the screen, if applicable |
| Display Criteria | Example: drop down box |
| Screen/Field Edits | Define edits on specific fields, if any. |
| Navigation | Describe how the user will access and use the screen; include previous screen and next screen as applicable |
| Error/Warning Messages | Define all error and warning messages that should be displayed and explain what action prompts these messages. |
| Buttons | Describe all screen buttons; include button name, its function and what happens when the button is used |

Attach a copy of the screen layout.

# Appendix B – Report Specifications

|  |  |
| --- | --- |
| Report Name | Description |
| Users | List the user area(s) that will receive the report. |
| Description | Describe the report. |
| Data Elements | List and describe the required data elements for the report |
| Data Selection Criteria | Define the selection criteria. |
| Sort Sequence | Document the sort sequence for the report |
| Control Breaks & Totals | Define totals and subtotals.  Define page breaks. |
| Logos | List all logos that will be needed. |
| Frequency | Indicate required frequency of the report |
| Distribution | Include location, number of copies, delivery method and mediums. |
| Retention | Include data retention period for reprints. |

Attach a copy of a report layout.

Template History:

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date | Author | Comments |
| 1.00 | 3/13/2008 | SEPG | Initial SDLC Rollout |
| 1.01 | 10/9/2008 | Kimbrough | Removed “auto update” setting on “Normal” style. |
| 1.02 | 11/5/08 | Manzari | PI #148 Fixed formatting problems |
| 1.03 | 5/8/09 | Stabler | PI #243 Added new tables to maintain traceability if the HLD is waived or existing used, so can trace back to functional/non-functional requirements. |
| 1.04 | 8/6/09 | Lezo | Data Classification standard label |
| 1.05 | 4/12/10 | Stabler | PI# 92 Peer Review Project. Added section with table for Code Quality Reviews – indicate which modules will undergo Peer Review |
| 2.00 | 4/12/10 | SEPG | Baselined |
| 2.01 | 4/20/11 | Manzari | Added approver table PIR 384 |
| SP v5 | 8/13/13 | Manzari | Added ILM guidance |