FUJ00090946

Document Title: TPS POL FS Summarisation HLD

Document Type: High Level Design

Release: S80

Abstract: This document describes the changes required in TPS (Transaction Processing Service) to summarise the transactions and generate Branch Ledger Entry Statements for the POL Financial System. TPS Host will produce flat files containing Branch Ledger Entry Statements each night for passing to POL FS.

Document Status: Approved

Originator & Dept: Sudhanshu Agrawal, Development Unit

Contributors:

Internal Distribution: Fujitsu Services Document Management

External Distribution:

Approval Authorities:

Name	Position	Signature	Date
Andy Kennedy	DU Design/Development Manager		
Gareth Jenkins	RASD Design Authority		

FUJ00090946

0 Document Control

0.1 Document History

Version No.	Date	Reason for Issue	Associated CP/PinICL
0.1	30/06/2004	First draft	
0.2	23/08/2004	Updated as per the review comments received. Added details of the new POL FS Summarisation processes.	
0.3	13/09/2004	Updated as per the review comments received.	
0.4	15/09/2004	Updated the reviewers' list as discussed with Document Management.	
0.5	22/09/2004	Updates as per the changes in POL FS AIS version 1.1. Minor updates as per the review comments.	CP3823
0.6	06/10/2004	Minor updates as per the review comments.	
1.0	01/12/2004	Updated reviewers/approvers list and sent for approval.	
1.1	02/12/2004	Updates as per the POL FS AIS version 1.2.	CP3843/44
1.2	05/01/2005	Updates to Opening Balances to split system generated cash figure and cash-in-pouches figure. Minor updates (mostly clarification) as per the review comments and PEAKs rasied.	CP3884
1.3	08/02/2005	Updated section 6.2.4 – Produce Opening Figures for handling Surplus Discrepancy and Loss System products.	
1.4	15/03/2004	Minor updates to POL FS Incomplete Summaries Report.	
2.0	19/08/2005	Removed Kevin Watson and Dave Johns from the approvers list, as they have moved off the project. Sent the document for approval.	

FUJ00090946

0.2 Review Details

Review Comments by :	
Review Comments to :	Sudhanshu Agrawal

Mandatory Review Authority	Name
Development Unit	Siva Gurunathan*, Wing Pang, Vaijayanti Gharpure, Duncan Brown, Bandna Tatter
DU Design Authority	Dave Johns, Rex Dixon, Pete Jobson*, Peter Ashdown*, Duncan Macdonald, Roger Barnes, Nasser Siddiqi
RASD Design Authority	Gareth Jenkins*
DU SV&I Manager	Debbie Richardson
DU Development Manager	Andy Kennedy
DU DeLT Manager	Julie Havard
CS System Support Centre Manager	Mik Peach*
CS Network Service Manager	Mark Ascott
CS Data Centre & Ops Service Manager	John Moran
Optional Review / Issued for Information	
DeLT	Miriam Bell
Host and DeLT Manager	David Harrison
Infrastructure/ Audit	Alan Holmes*, Andy Scott

(*) = Reviewers that returned comments

0.3 Associated documents

	Reference	Ver.	Date	Title	Source
[R1]	EA/IFS/006			Pathway to POL MIS AIS	Fujitsu Services
[R2]	EA/IFS/005			Horizon to POL Client Transmission Summaries AIS	Fujitsu Services
[R3]	EA/IFS/003			POL FS AIS	Prism/ Xansa
[R4]	TI/IFS/008			Horizon to Post Office Technical Interface Specification	Fujitsu Services
[R5]	EA/HLD/010			IMPACT Release 3: Agents High	Fujitsu

FUJ00090946

				Level Design	Services
[R6]	TI/MAN/002			TPS Host Support Guide	Fujitsu Services
[R7]	TI/DES/002			TPS High Level Design	Fujitsu Services
[R8]	BD/CDE/008			PO Ltd Financial Systems Release 3 Conceptual Design	Post Office Ltd / Prism
[R9]	EA/DPR/004			Impact Release 3 Design Proposal	Fujitsu Services
[R10]	TD/ION/005			FTMS Configurations for TPS and PO Ltd TIP Links	Fujitsu Services
[R11]	AD/DES/047			TPS Tables and Mappings for CSR+	Fujitsu Services
[R12]	DW/IFS/021			TPS to DWh (Fujitsu Services) Application Interface Specification	Fujitsu Services
[R13]	RD/IFS/018			RDMC - TPS Application Data Interface Specification	Fujitsu Services
[R14]	EA/HLD/009			TPS HR SAP Summarisation & Transaction Correction HLD	Fujitsu Services
[R15]	EA/HLD/008			IMPACT Release 3 Migration HLD	Fujitsu Services
[R16]	PA/PER/033			Horizon Capacity Management and Business Volumes	Fujitsu Services

Note: Unless a specific Version and Date is referred to above, reference should be made to the current (Approved) Version of the document.

0.4 Abbreviations/Definitions

ADC	Advanced Distribution Centre: Used as an abbreviation on the Horizon desktop for Remittances to and from SAP ADS
Article	Article is a SAP IS related terminology for product or service
AIS	Application Interface Specification
APS	Automated Payments System (Fujitsu Services)
BLE	Branch Ledger Entry
BP	Balance Period
CAP	Cash Account Period
COFA	Chart of Accounts
CTT	Counter Transaction Timings
DCP/DCS	Debit Card Project/ Debit Card Service

FUJ00090946

DWh	Data Warehouse (Fujitsu Services)
EDSC/SSC	System Support Centre
EPOSS	Electronic Point of Sale Service (Fujitsu Services)
FAD	Financial Accounts Division (FAD Code)
I/f	Interface
Impact	The programme within Post Office Ltd which is making change to Improve the accounting processes.
Movement Type	Movement type on POL FS
LFS	Logistics Feeder Service
MSU	Management Support Unit (formerly Business Support Unit)
NWB	Network Banking
OMDB	Operational Management Database
ONCH	Overnight Cash Holding
POL	Post Office Limited (formerly POCL)
POL FS	Post Office Ltd's Financial System
RDMC	Reference Data Management Centre
RDS	Reference Data System
SAP ADS	SAP Advanced Distribution System
TIP	Transaction Information Processing (POL)
TPS	Transaction Processing Service (Fujitsu Services)

0.5 Changes in this version

Version 0.2

- Updated various sections as per the review comments received.
- Added details of POL FS Summarisation Processes.

Version 0.3

Updates to various sections as per the review comments received.

Version 0.4

Updated the reviewers' list as discussed with Document Management.

Version 0.5

- CP3823 - Updates as per the changes to POL FS AIS version 1.1.
- Minor updates (mostly clarification) to various sections as per the review comments.
- Added a section on SLA data (on POL FS file delivery) to be sent to Fujitsu Data Warehouse.
- Added the locally defined mappings for Pre-migration (Appendix E) and Opening Figures (Appendix F) Summaries.

FUJ00090946

Version 0.6

- Overall view on Harvester Exception tables needs to be recreated as new a column is added to Harvester Exception tables.
- Removed the Financial Transaction flag in the Harvester Exceptions table, as it is difficult for the TPS Harvester to populate it.
- Removed '9999' from the Line Number in the spreadsheet for TPS_OPENFIG_CAC_PROD_MAPPINGS table.
- Updated the spreadsheet for POL_FS_MAPPINGS_AT_S60. Rows for articles 999999 and 999998 are not to be loaded in the table.
- Empty POL FS Summary files will be deleted. File sequence 201-264 will be used for Opening Figures (BLCR2) and 301-364 for Pre-migration Summaries (BLCR3).

Version 1.0

- As required, updated the reviewers/approvers list and sent for approval

Version 1.1

- Updated various sections as per the changes POL FS AIS version 1.2 (CP3843/44).
- Updated sections 6.8.3 and 7.1 to stop producing POL FS Files as per the S60 format from migration point 25 onwards.
- PEAK111264 - Updated the document to clarify that 65th partition of Initial Summary table should be truncated by TPSC209.
- Removed the hard coded filter on the Cash Account Line Numbers from the "Generate Opening Balances" SQL, as the filter is automatically applied via the CA Line Numbers present in the TPS_OPENFIG_CAC_PROD_MAPPINGS table.
- Updated the TPS_POL_FS_MAPPINGS_AT_S60 spreadsheet in the Appendix E with the "Article Type" values.

Version 1.2

- Updated the TPS_POL_FS_MAPPINGS_AT_S60 spreadsheet in Appendix E with the "Article Type" values. Added the RDDS equivalent of TPS tables containing reference data in the section 6.1.1.
- CP3884 - Updated section 6.2.4 to stop suppressing the opening figure for Product 1 (Cash). Also, added Suspense Product 5610 (Cash in Pouches) to the TPS_OPENFIG_CAC_PROD_MAPPINGS spreadsheet in Appendix F.
- Minor updates (clarification only) to section 6.7 as per the review comments.
- PC114105 – Updated Appendix D to clarify that Amount and Quantity on the Incomplete Summary report should be written with appropriate sign.

FUJ00090946

- PC113637 – Updated section 6.3.2 to make use of overall view on TMS_HARVESTER_EXCEPTIONS (partitions 1 to 64) tables.
- PC112521 – Updated section 6.2.1 to make use of QUANTITY column for the Bureau transactions of APPLICATION_TYPE = 'BDC2'.

Version 1.3

- PC115380 - Removed products 145 and 222 as these are also harvested via Stock Holdings route.
- PC115465 – While generating Opening Figures, if MAP.QUANTITY flag is set to 'QV', then select Amount in Pence as the Sum of Quantity.
- PC115574 – As the products Cash (1) Cash-in-Pouches are selected from the Stock Holdings table and Cash Accounts table respectively, but map to the same Article, the SQL (in section 6.2.4.2) needs a further level of summarisation, i.e., by Branch, Trading Date and Article Id.

Version 1.4

- PC117536 – Update SQL in POL FS Incomplete Summaries Report to extract only those records where PROCESSED flag is NOT set to 'Y'. Also, clarify by adding a page-footer that this is delta position for the date, not the cumulative position of POL FS Incomplete Summaries.

0.6 Changes forecast

None.

FUJ00090946

0.7 Table of Contents

FUJ00090946

FUJ00090946

FUJ00090946

1. Introduction

This document describes the changes in TPS Host to summarise the transactions and generate Branch Ledger Entry Statements (BLE) for the POL Financial System.

A set of fixed format text files will be generated each night for passing to POL FS. The structure of the file is basically the same as for S60, however the detailed record formats have changed. Also, instead of building up the summaries at the counter and passing them straight through to the file generation function, the summarisation will now take place at the TPS Host.

The interface between TPS Host and POL FS is defined in [R3], which includes the file and record format, and transferring process for the file(s) from TPS Host to POL FS machine.

The detailed requirements and design proposal for the PO Ltd Financial Systems Release 3 System is available in the Conceptual Design [R8] and Design Proposal [R9].

This document is an internal Fujitsu Services document. The level of detail in this document is intended to act as a baseline to Fujitsu Services developers and testers.

2. Scope

The document covers the changes (at high level) needed in TPS Host to generate and forward Branch Ledger Entry statements to the new POL Financial System. This document is a delta HLD for the S80 changes and will be later merged into TPS HLD [R7].

The document does not cover the details of Agent Harvesters, which populate the transactions and events into TPS database. The details of Agent Harvester are present in TPS Agents High Level Design [R5] and TPS Tables and Mappings [R11].

Further, the HLD for HR SAP summarisation and Transaction Correction will be initially covered by a separate document [R14].

This document does not provide the details of the file transfer process. The files will be copied to an NFS share and picked up by the POL FS system.

3. Assumptions

It is assumed that the Agent Harvesters have successfully populated the Transactions tables in TPS Host before the host processing starts. Also, the Agents harvest either the full day's worth of transactions for a Branch and Trading Date or no transactions.

Further, the Agents harvest all transactions for a Branch and Trading Date into the same partition of TPS Transaction tables. For example, all EPOSS, OBCS, APS,

FUJ00090946

NWB, DCS and Bureau transaction for the Branch 234543 and Trading Date 22-Apr-2004 will be harvested into partition 1 of these Transaction tables.

For the reference data available in POL FS Mapping tables at Migration Point 25 (see section 6.8), it is assumed that

- there is a mapping available for each Product and Transaction Mode where the transactions are required to be sent at Article Mode (Summarisation = 'P') or Unsummarised (Summarisation = 'T'). Otherwise, these Products and Modes will not be picked up by the summarisation processes. A check in RDDS ensures that this assumption is valid.
- there is a default mapping available for each Product where the transactions are required summarised (Summarisation = 'S'). This default mapping will have a Transaction Mode value of zero. If a product has a default mapping (Transaction Mode = 0), no mappings for specific Transaction Modes can exist for the Product.
- the mappings do not change during a business day. That is, TPS doesn't need to apply different mappings to transactions taking place at different times of a day. This is because the Start Date (and End Date) present in the mapping tables will be compared with the Trading Date (at 08:00 hours) present in the transactions to derive the appropriate mappings.
- there is no overlap between the End and Start Dates.

To generate pre-migration (Migration Point 30) POL FS Summaries, it is assumed that all products required in the summarisation are present in the locally defined mappings present in TPS_POL_FS_MAPPINGS_AT_S60 table.

To generate new opening figures for all Foreign Exchange (Bureau) products, it is assumed that TPS Harvester has populated a new column PURCHASED_QUANTITY in the TMS_RX_STOCK_HOLDINGS table with <AdditionalData.BDC.PQty:> attribute from the Opening Figures message. For non-Bureau products this column value must be set to NULL.

FUJ00090946

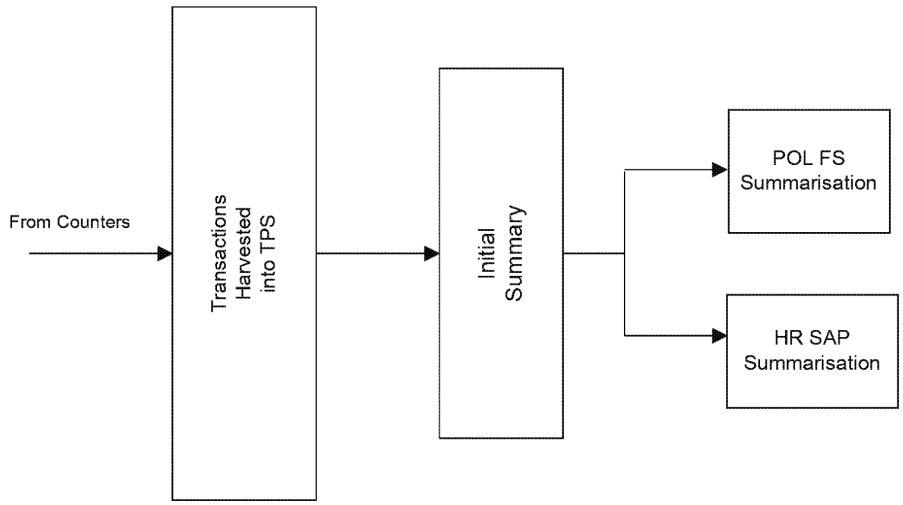
4. Overview

At S60, the transaction summarisation was done at the Branch and the summaries were harvested into TPS for passing onto POL FS.

However, at S80 it is proposed that a new transaction summarisation process will run in TPS Host as part of the overnight processing. The summaries will then be passed onto POL FS.

Summarisation will be a two-stage process:

- Initial Summarisation of Product and Mode
- POL FS Summarisation and HR SAP Summarisation



In the initial summarisation of Product and Mode, the transactions will be summarised by Product / Mode combinations such that for each Branch and Trading Day there is a single Summary Record for each combination of Product / Mode that has taken place during the Trading Day. Some transactions, e.g., CASH REM IN, CASH REM OUT, as defined by 'Article Mode' Reference Data, will be passed unsummarised

The data from initial summarisation will be used for a further set of summarisation - POL FS Summarisation and HR SAP Summarisation.

POL FS summarisation, based on Reference Data, will map the Product / Mode summaries for each Branch / Trading Day onto the data to be passed to the relevant POL FS Articles and Accounts. POL FS needs the data for all branches.

HR SAP summarisation, again based on Reference Data, will map the Product / Mode summaries for each Branch / Trading Day onto the data required for HR SAP based on CTT Numbers. HR SAP summarisation is not required for the Directly Managed

FUJ00090946

Branches (about 600 in total). The details of HR SAP summarisation are in separate document [R14].

The summarised data for POL FS will be read from the POL FS Summary tables and written to flat data files as per the record formats present in POL FS AIS [R3]. A data file may contain one or more sub-files (empty files contain File Header and Trailer only, no sub-files, and will be deleted). There will be one sub-file per Branch and Trading Day. The data files will be created in a predefined host directory. There will be up to 64 data files (one per normal table partition) per day from the normal POL FS Summary tables and 1 data file per day from the POL FS Incomplete Summary table[1].

A check will be made so that all data in a sub-file has a net value of zero. If this is not the case, then an operational exception will be raised and the summaries held back until a correcting transaction is generated[2] to enable the sub-file to balance correctly.

One of the reasons for the sub-file total not balancing to zero is that some transactions have failed harvesting due to a database check constraint failure. In such cases, the transactions will be repaired before they are passed through. This means, these transactions will not be available immediately for summarisation.

In order to handle the case where the transactions are known to be missing (i.e., they have failed harvesting), there is no point in attempting to produce a sub-file for passing to POL FS since it will be incomplete. Such cases will be detected by examining the TPS Harvester Exceptions Archive table. If there is a harvester pending exception present in this table, the related summaries will be held back for processing on a subsequent day when the exceptions have been repaired.

FTMS transfer will not be used for the data files transfer to the POL FS system. Instead, the files will be copied to an NFS share and picked up by the POL FS system. For Auditing the data files, a Unix link will be created in the Audit directory.

To record the delivery of the POL FS files, the date and time of file copy to NFS share will be recorded into TPS database. This date and time will be used in file delivery information generated for Data Warehouse (Fujitsu Service).

The summarised data will be stored in a set of transient tables and will either be written to POL FS file (if the sub-file total balances to zero) or saved in a separate table (if the sub-file total does not balance to zero). Once processed, the summarised data will be deleted from the database by the TPS End of Day process.

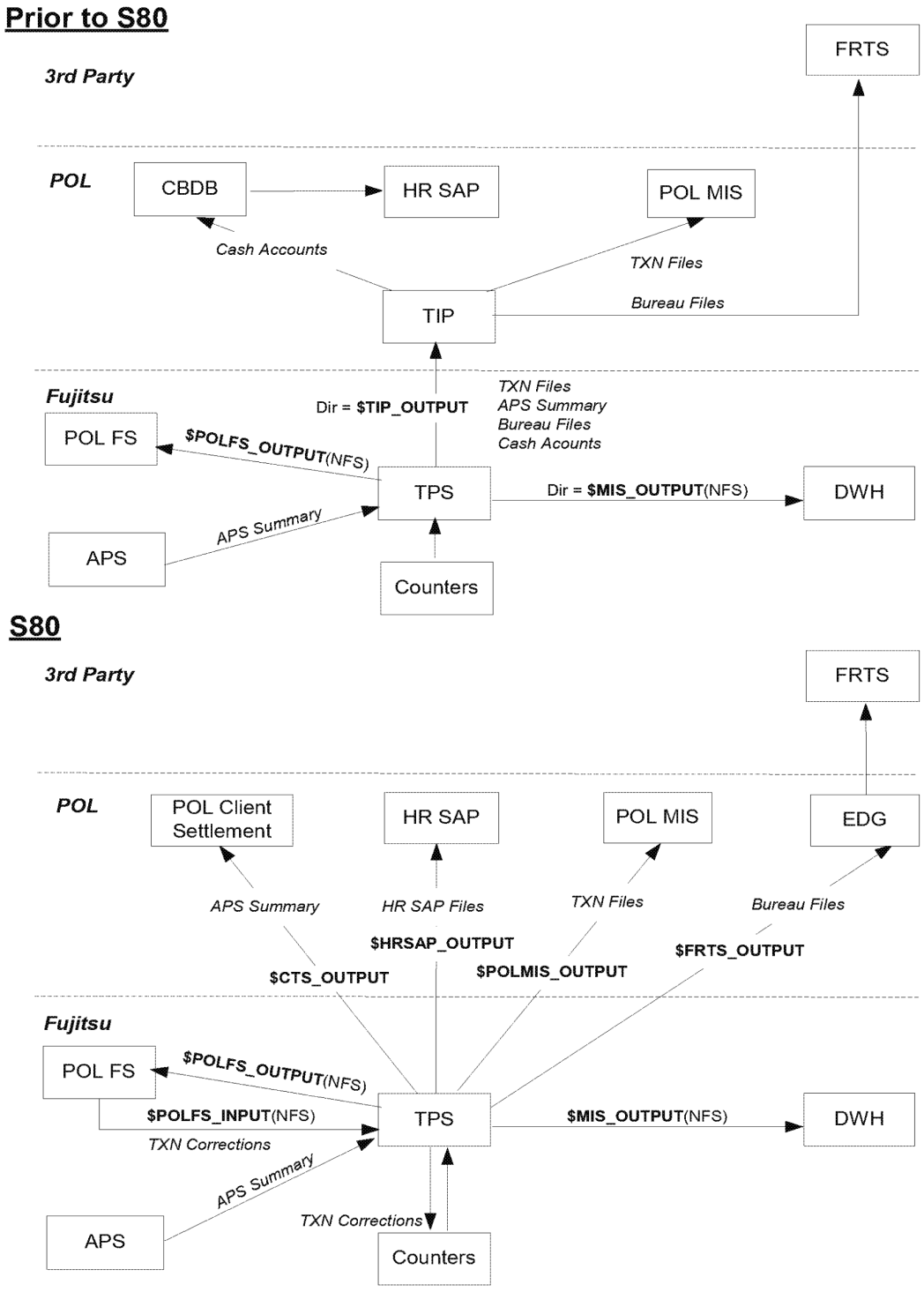
---

[1] POL FS Incomplete Summary table will contain held back summaries where the total for a Branch and Trading Date did not balance to zero on a previous day due to harvester exceptions or any other reason, but now the transactions have been corrected/repaired, so the summaries balance to zero.

[2] The balancing transaction will automatically be generated if the discrepancy was due to harvester exceptions. Otherwise, SSC have to manually insert a balancing transaction into POL FS Incomplete Summary table. It is expected that a manual insert of balancing transaction will be very rare.

FUJ00090946

5. Architecture



FUJ00090946

6. Detailed Design

This document mainly concentrates on the POL FS Summarisation. The other changes (POL MIS Feed, HR SAP Feed, etc.) to TPS host in S80 release are covered by [R14].

The design for POL FS Summarisation breaks down into the following:

1. Creating the Oracle objects to store data for POL FS Summarisation
   - Creating the Oracle objects to store Reference Data
   - Adding new columns to existing Transaction Tables
   - Creating the Oracle objects to store Initial Transaction Summaries
   - Creating the Oracle objects to store POL FS Summaries
2. Transaction Summarisation Processes
   - Initial Summarisation Process
   - POL FS Summarisation Process
3. Producing the POL FS Summary Data and EOT Files
   - Producing the Data File from POL FS Incomplete Summaries Table
   - Producing the Data Files from Normal POL FS Summary Tables
   - Producing the EOT File
   - Host Directories and File Names for POL FS Summary and EOT Files
   - Reporting Incomplete POL FS Summaries
4. Handling Harvester Exceptions due to Check Constraint Failure(s)
5. Housekeeping the Transaction and Summary Data Tables at TPS End of Day
6. POL FS Summary File Rejection Handling and SLA Reporting
7. TPS Host Maestro Schedule Changes
8. Migration from Counter Summarisation and Providing Opening Figures

6.1 Creating the Oracle objects to store data for POL FS Summarisation

New tables will be created in TPS database to store the reference data and transaction summaries for POL FS. For transaction summaries, there will be two sets of tables to store the current and previous day's transaction summaries. These table sets will be used in the same way as the current TPS Transaction tables are used. That is, one set will store the current day's data and the other set will store previous day's data. Each set will have 64 normal partitions. The data in 64 normal partitions will be accessed through views. Details of accessing through views are in TPS HLD [R7]. There will

FUJ00090946

be a 65th partition of Initial Summaries to store the summaries generated using repaired transactions.

The advantage of keeping the existing partitioning mechanism is that no changes are required to the TPS Transaction tables or Agent Harvester. Further, having two sets of tables for the summarised data including the intermediate summaries will help SSC in analysing and resolving any data discrepancies found in the yesterday's data sent to POL FS without referring to the backed up or archived data.

6.1.1 Creating the Oracle objects to Store Reference Data

A set of tables will be created in TPS database to store the required reference data for POL FS summarisation.

TPS will copy the required reference data from the RDDS database on a daily basis. The data copy will be over a database link from TPS to RDDS database. Refer to [R13] for the details of TPS to RDDS Interface.

TPS_POL_FS_ARTICLES (POL_FS_ARTICLES in RDDS)

The logical layout of the table containing POL FS Articles is as follows:

Data Attribute	Description
Article Id	POL FS Article All Horizon Products will map to a POL FS Article. This may be a dummy article to facilitate the consistent mapping of all Horizon Products. Articles will be created for items such as Cash in hand, cheques etc that will not impact an article in POL FS but actually map to a GL or Agent Account instead.
Description	Description of the Article. Included for information and diagnostics.
Article Type	Used by POL FS end of interface for control over Idocs and other postings. The values will be defined by reference data.
Default Product Id (Positive Correction)	The default product that transaction corrections for the Article are to be converted to. Optional - required for each Article for which transaction correction may be applied.
Default Product (Negative Correction)	The default product that transaction corrections for the Article are to be converted to. Optional - required for each Article for which transaction correction may be applied.
Article Summarisation	Indicates whether transactions are to be summarised at article level rather than mode or transaction level. Vales are 'Y' or 'N'.
Quantity	Defines the use of quantity Values are 'CT' Transaction Count 'CQ' Item Count 'NA' Not Applicable 'QV' Value

The attributes like "Article Type" which are required for POL FS Summaries will also be present in the de-normalised table TPS_POL_FS_MAPPINGS. The attributes like Default Product Ids will be used in Transaction Correction. The details of Transaction Correction processes are present in TPS HR SAP Summarisation & Transaction Correction HLD [R14].

FUJ00090946

TPS_POL_FS_ACCOUNTS (POL_FS_ACCOUNTS in RDDS)

The logical layout of the table containing POL FS Accounts is as follows:

Data Attribute	Description
Account	POL FS Account
Description	Description of the Account. Included for information and Diagnostics.
Ledger	Defines which Ledger the POL FS account belongs to. Values are 'G' General Ledger 'C' Client Ledger 'A' Agents ledger
Settlement	Defines which data is used to settle with the client Values are 'P' Post Office Data 'C' Client Data 'E' Estimated Data

TPS_POL_FS_MAPPINGS (POL_FS_MAPPINGS in RDDS)

The logical layout of the table containing Horizon Product and Transaction Mode to POL FS Article and Account mappings is as follows:

Data Attribute	Description
Product Id	Horizon Item ID – the item id for the product being transacted.
Transaction Mode Id	Horizon transaction mode. Transaction modes differentiate transactions such as sales, reversals, adjustments, transaction corrections, remittances in and out etc. Where the products are required summarised, the Transaction Mode Id will be set to 0.
Start Date	Date when effective.
End Date	Date when ceases to be effective.
Article Id	POL FS Article All Horizon Products will map to a POL FS Article. This may be a dummy article to facilitate the consistent mapping of all Horizon Products. Articles will be created for items such as Cash in hand, cheques etc that will not impact an article in POL FS but actually map to a GL or Agent Account instead.
Article Type	.Used by POL FS end of interface for control over Idocs and other postings. The values will be defined by reference data
Movement Type	POL FS Movement type This is only relevant for stocked products and allows different movement types such as REM in, REM out, and adjustments to be shown differently in SAP. There will be a consistent mapping between certain modes to movement types but these will only be valid for particular Articles.
Account	POL FS account number This is the account number within the Ledger defined above. This field is optional in the file because records that relate only to stock movements with no financial impact will not map to an account in any ledger.

FUJ00090946

Quantity	Defines use of Quantity field in reference to the Horizon product/ POL FS Account NA = Not applicable – hence field blank CT = Count number of transactions (applicable to services) CQ = Count number of items of product sold (applicable to stocked products) QV = The Quantity field is populated from the value in Horizon. This is used for products similar to Other Stamps, which are value managed in Horizon but do have a real accounting stock value. Therefore POL FS is going to record the value in stock as the quantity.
Settlement	Defines which data is used to settle with the client P = Post office data C = Client Data E = Estimated data This is required as there will be a different indicator used within POL FS to differentiate these data types. (Special GL indicator). This is optional as it is only relevant to client products and transactions that need to feed the Client ledger.
Ledger	Defines which Ledger the POL FS account belongs in G = General Ledger C = Client Ledger (AP) A = Agents Ledger (AR) POL FS uses three different ledgers and needs to know which ledger each account number is part of. This field is optional in the file because records that relate only to stock movements with no financial impact will not map to an account in any ledger.
Summarisation	Indicator showing whether the following is valid: S = Summary Data at branch/article/day P = Summarised by "Article Mode" indicated. T = Transaction Data – not summarised at all Transaction Data is anticipated to be remittances of cash, and Transaction Corrections.
Reference	Defines use of Reference field. NA = Not applicable PC = Pouch Bar Code TC = Transaction correction reference number Summarised data will never have a reference. Transaction data will often, but not always, have a reference

The contents of the above table will be derived by RDDS and made available to TPS. The contents will be based on RDDS tables like POL FS Article, POL FS Account, Item Article Mappings and Article Mode Mappings. The corresponding table in RDDS will contain all current and future POL FS mappings to products and transaction modes and a 50 day history of past mappings.

To find the latest mapping available for a Product and Transaction Mode, the Start Date and End Date present in the mappings table will be compared with the Trading Date present in the Transaction. For example,

Start Date <= Trading Date + 08:00 hours < End Date (or End Date set to null)

As suggested by RDDS, a Trading Day starts at 08:00 hours. So, 08:00 hours has been added to the Trading Date while doing the above comparison.

FUJ00090946

TPS_POL_FS_MAPPINGS_AT_S60 (Locally defined in TPS)

This table will store locally defined mappings for Cash and Near Cash products only and will be based on POL FS mappings at S60. The table structure will be same as TPS_POL_FS_MAPPINGS table.

Having a separate table for S60 mappings will keep the locally defined mappings separate from those copied from RDDS. Also, it helps if some column values are not required, hence a relaxed set of check constraints is implemented for these mappings.

These mappings will be used for POL FS Summaries generated by TPSC292 for CAP <= Final CBDB CAP.

TPS_TRANS_MODE_CONVERSIONS (POCL_TRANS_MODE_CONVERSIONS in RDDS)

The logical layout of the table containing numeric Post Office transaction modes to alphabetic Fujitsu Services transaction modes is as follows:

Data Attribute	Description
POL Transaction Mode Type Code	Post Office transaction mode (numeric). For example, 2 Remit In - Supplies Division 3 Remit Out - Supplies Division 4 Revaluation - Uprating 5 Revaluation - Downrating
Pathway Transaction Mode Type Code	Horizon transaction mode (alphabetic). For example, RISD Remit In - Supplies Division ROSD Remit Out - Supplies Division RU Revaluation - Uprating RD Revaluation - Downrating

TPS_POL_FS_EXCLUDED_MODES (POL_FS_EXCLUDED_MODES in RDDS)

The logical layout of the table containing Transaction Mode to be excluded from POL FS feed is as follows:

Data Attribute	Description
Transaction Mode Id	Horizon transaction mode to excluded from POL FS feed. Transaction modes differentiate transactions such as sales, reversals, adjustments, transaction corrections, remittances in and out etc.

TPS_OPENFIG_CAC_PROD_MAPPINGS (Locally defined in TPS)

The logical layout of the table containing locally defined Cash Account Line to Product Id mappings for Discrepancy and Suspense Products used in generating the Opening Figures for POL FS Summaries is as follows:

Data Attribute	Description
Line Number	Cash Account Line Number
Product Id	Horizon Item ID – the item id for the product being transacted.

FUJ00090946

Accounting Sense	Multiply the Cash Account Line value by 1 if accounting sense is "+1" or -1 if accounting sense is "-1".

The details of synonyms, roles and grants on the above tables are in Appendix A to C of this document.

6.1.2 Adding new columns to existing Transaction Tables

Three new fields will be added to the existing **EPOSS Transaction** (only) tables to store the Transaction Ids need to be passed to POL FS. These Ids will be passed only for unsummarised transactions sent in the POL FS Summaries.

Data Attribute	Description
Pouch Id	Pouch Identifier (this field may be needed if the information is being sent unsummarised) Controlled by reference data. This is the value of EPOSSTransaction.BlackBoxData.PouchId. To be harvested into a new column in the EPOSS interface table.
TC Reference	Transaction Correction Reference (this field may be needed if the information is being sent unsummarised) Controlled by reference data. This is the value of EPOSSTransaction.BlackBoxData.Ref. To be harvested into a new column in the EPOSS interface table.
Client Reference	Client Reference number (supplied by client) This will only be used for the actioned Transaction Corrections being passed back to POL FS against the client products. NOT controlled by reference data. This is the value of EPOSSTransaction.BlackBoxData.AddRef. To be harvested into a new column in the EPOSS Transaction table.

Details of other additions to the existing tables are present in Appendix A of this document.

6.1.3 Creating the Oracle objects to Store Initial Transaction Summaries

Two new sets of tables will be created in TPS database to store the initial transaction summaries. One set will hold the current day's data and one will hold the previous day's data.

TPS host will need to run multiple instances of the process in parallel, viz., eight instances each processing eight table partitions, to speed up the two stage summarisation process and produce POL FS files within the specified time period. Also, after processing each partition, the process will insert a record in the Process Control table indicating its run status. This will show whether a table partition has been processed successfully, which helps in reducing the amount of data to be reprocessed in case of a restart after failure, as the restart doesn't need to go through

FUJ00090946

the partitions which has marked as "successfully processed" in the Process Control table. Moreover, running parallel processes complies with the current TPS Host setup.

To assist parallel processing there will be 64 tables for Initial Summaries for the data summarised from normal transaction tables. The setup for Initial Summaries tables will be same as that for the TPS Transaction tables.

There will be a 65th partition of the Initial Summary table to store the summaries generated from repaired transactions (in 65th partition). There is no need to have an archive table associated with the 65th partition of the Initial Summary table, as the partition contains intermediate data which can be regenerated from the archived repaired transactions, if required.

A separate instance of Initial Summarisation process will process the repaired transactions in 65th partition.

The logical layout of Initial Summary table is as follows:

Data Attribute	Description
Trading Date	The Trading day during which the transaction took place. EOD Marker defines the end of a Trading Day. N.B. The time the EOD Marker is written varies from Branch to Branch.
Group Id	Group Id (FAD Code without the check sum) for the Branch.
Product Id	The Horizon Product Identifier
Transaction Mode Id	Horizon Mode the product was transacted in, e.g. Sell, Rem In, Revaluation, etc.
Total Transaction Amount	Sum of the amount field in the source tables in £ (sterling).
Total Transaction Quantity	Sum of the quantity field in the source tables i.e. number of items transacted.
Transaction Count	Count of transactions in the total
Pouch Id	Pouch Identifier (this field may be needed if the information is being sent unsummarised) Controlled by reference data. This is the value of EPOSSTransaction.BlackBoxData.PouchId. To be harvested into a new column in the EPOSS interface table.
TC Reference	Transaction Correction Reference (this field may be needed if the information is being sent unsummarised) Controlled by reference data. This is the value of EPOSSTransaction.BlackBoxData.Ref. To be harvested into a new column in the EPOSS interface table.
Client Reference	Client Reference number (supplied by client) This should only be used for the actioned Transaction Corrections being passed back to POL FS against the client products. However, no validation will be done by TPS Host and any value present will be included. NOT controlled by reference data. This is the value of EPOSSTransaction.BlackBoxData.AddRef. To be harvested into a new column in the EPOSS Transaction table.

FUJ00090946

Cash Account Period	Cash Accounting Week that comes as a part of Brach Cash Account. This is required during the migration period only.
TPS Insert Date	Date and Time of initial transaction summarisation by TPS host. Defaulted to Oracle SYSDATE.

Partitions 1 to 64 will be accessed via a set of views similar to those for the transaction tables. The 65th partition will be accessed via a public synonym. The physical layout of the Initial Summary table is present in Appendix A.

6.1.4 Creating the Oracle objects to Store POL FS Summaries

Two new sets of tables will be created in TPS database to store the POL FS summaries. One set will hold the current day's data and one will hold the previous day's data.

TPS host will run multiple instances of the process in parallel, viz., eight instances each processing eight initial summary table partitions, to speed up POL FS summarisation process and produce the files within the specified time period.

To assist parallel processing there will be 64 tables for POL FS Summaries for the data summarised from the Initial Summary tables. The setup for POL FS tables will be same as that for the Initial Summary tables.

There will be a separate table to store the POL FS Incomplete Summaries. This table will store the POL FS summaries where the sub-file total did not balance to zero. Also, this table will store the POL FS summaries generated from repaired transactions (in 65th partition). The POL FS Incomplete Summaries table will hold data as long as required, i.e., until the summaries have been corrected and sent to POL FS. The table structure for POL FS Incomplete Summaries will be same as that for POL FS Summaries table except a flag to indicate that the summaries have now balanced and sent to POL FS.

The logical layout of POL FS Summary table is as follows:

Data Attribute	Description
Trading Date	The Trading day during which the transaction took place. EOD Marker defines the end of a Trading Day. N.B. The time the EOD Marker is written varies from Branch to Branch.
Group Id	Group Id (FAD Code without the check sum) for the Branch.
Transaction Mode Id	Horizon Mode the product was transacted in, e.g. Sell, Rem In, Revaluation, etc.
Transaction Mode Acronym	This is the acronym (e.g. SC, DDP, DDN, RIAD, ROAD, etc.) for Transaction Mode Id (e.g. 1, 17, 19, 23, 25, etc.). The summary records sent to POL FS will contain Transaction Mode Acronym.
Article Id	POL FS Article Identifier.
Movement Type	Movement Type indicates Movement type on POL FS.
Article Type	Used by POL FS end of interface for control over Idocs and other postings. The values will be defined by reference data. The use of this will be defined by reference data.
Settlement Type	Indicator showing whether settlement is based on POL or client or estimate data.
Ledger	POL FS Ledger indicator.

FUJ00090946

Account	Account Group identifier.
Account Value	Total daily value of Account in £ (sterling) Format = "999999999v99" This will be converted to pence when written to POL FS data file.
Account Quantity	Quantity relating to Account. The exact use of this field will be defined by the reference data. The options are: • Not Applicable (send spaces). • A count of number of transaction that have taken place. • A count of number of items sold. Depending upon the "Quantity" flag in TPS_POL_FS_MAPPINGS table, appropriate value from the initial summaries will be copied into this column.
Account Reference Id	Transaction Reference, if any. The field will either have Pouch Id or TC Reference. Depending upon the "Reference" flag in TPS_POL_FS_MAPPINGS table, appropriate value from the initial summaries will be copied into this column.
Client Reference	Client Reference number (supplied by client) This will only be used for the actioned Transactions Correction being passed back to POL FS against the client products.
Transaction Count	Count of transactions in the sum - used for SLA reporting to Data Warehouse.
Subfile_Id_Flag	A flag indicating the whether the summaries are for the Cash Account Period > Final CBDB CAP. Also, whether the summaries represent Opening Figures. Set to "BLCR1" for Summaries with CAP > Final CBDB CAP "BLCR2" for Summaries with Opening Figures on receipt of Final CBDB CAP. "BLCR3" for Summaries with CAP <= Final CBDB CAP This flag will be used around the migration period to identify the sub-file id, viz., BLCR1, BLCR2 or BLCR3.
TPS Insert Date	Date and Time of POL FS summarisation by TPS host. Defaulted to Oracle SYSDATE.

The POL FS Summary tables (partitions 1 to 64) will be accessed via a set of views similar to those for the TPS Transaction tables. The POL FS Incomplete Summary table will be accessed via a public synonym.

The physical layout of the POL FS Summary table is present in Appendix A.

FUJ00090946

6.2 Transaction Summarisation Processes

The transaction summarisation will be done in two stages:

- Initial Summarisation
- POL FS Summarisation

In the Initial Summarisation, i.e. the first stage, the transactions will be summarised by Branch, Trading Day, Horizon Product and Transaction Mode (and Cash Account Period during the migration). However, some transactions, which are required unsummarised, will have their Account Reference Ids and/or Client Reference included in the initial summary. The data from initial summarisation will also be used for HR SAP Summarisation (refer to [R14] for details).

In the POL FS Summarisation, i.e. the second stage, the Horizon Products and Transaction Modes will be summarised into POL FS Articles and Accounts.

The Product and Transaction Mode to POL FS Article and Accounts mappings will be available via the reference data in TPS_POL_FS_MAPPINGS table as described in the section 6.1.1. The summarisation can either be done in the SQL or in the C program. The recommendation is to use SQL, so that it is simpler, and easy to maintain.

TPS host will run multiple instances of the summarisation process in parallel, viz., eight instances each processing eight table partitions. There will be separate instance to process the repaired transactions in the 65^{th} partition.

Running multiple instances of a process will be scheduled by Maestro. There will be no complexity built into the application or code. The advantage of using multiple partitions and processes is breaking down a large chunk of work into smaller pieces which can have their own commit units. For example, if 63 of 64 processes succeeded then each partition will be marked as "successfully processed" in the Process Control table. The restart will then need to reprocess only one partition.

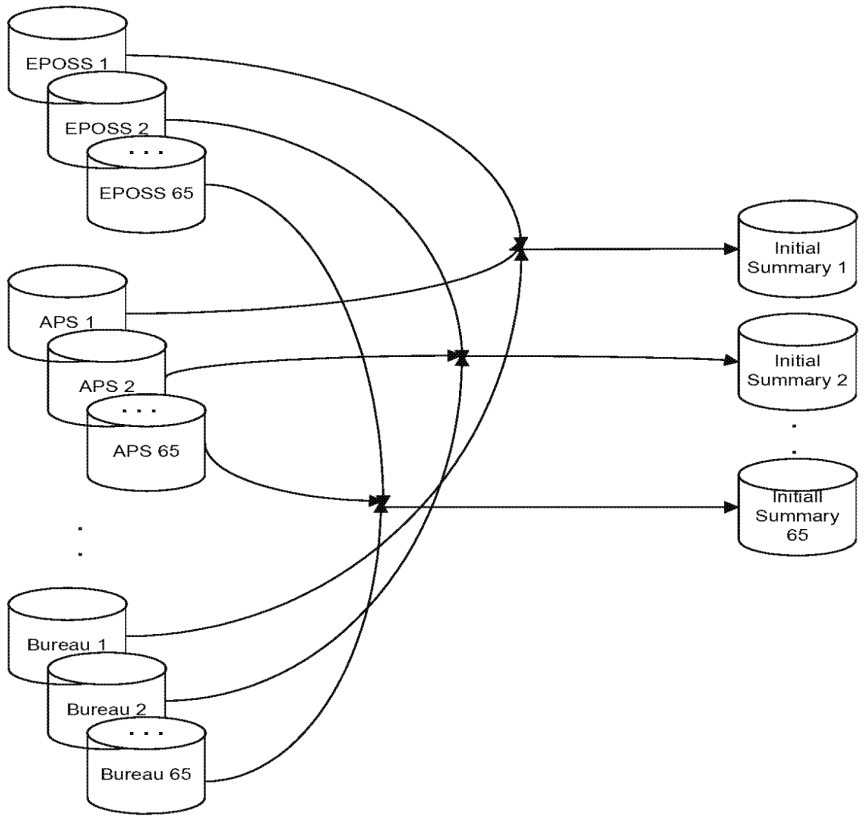
Also, the partitioning logic for Transaction tables (64 separate table; partition key value is not stored explicitly), which helps in parallel harvesting, smaller sort for POL MIS files and parallel writing to multiple data files, will be used throughout the application. Otherwise, there will be Oracle deadlocks with any kind of parallel processing.

FUJ00090946

6.2.1 Initial Summarisation Process

In the Initial Summarisation, each partition of transaction tables will be summarised by Branch, Trading Day, Horizon Product, Transaction Mode and CAP. However, some transactions, which are required unsummarised, will have their Account Reference Ids and/or Client Reference included in the Initial Summaries. The "Summarisation" flag in the TPS_POL_FS_MAPPINGS table (see section 6.1.1) will be used to decide whether a Product and Transaction Mode are required unsummarised.

The data from initial summarisation will also be used for HR SAP Summarisation (see [R14] for details).



There will be multiple instances of the initial summarisation process running in parallel, viz., eight instances each processing eight normal table partitions. There will be a separate instance to process the repaired transactions in the 65th partition.

The initial summarisation by Branch, Trading Day, Horizon Product, Transaction Mode and Cash Account Period (Cash Account Period has been included for the filter

FUJ00090946

to be used during migration) will be done in the SQL with appropriate "WHERE" clause to send the required Product and Transaction Mode unsummarised.

The rules for generating Initial Summaries are:

- Merge (UNION ALL) transactions from one partition of all TPS Transaction tables and select Trading_Date, Group_Id, Prod_Id, Transaction_Mode_Id, Cash Account Period, Sum of Quantity, Sum of Amount, Count of Transactions. For example, merge the transactions from TMS_RX_EPOSS_TRANSACTIONS_1, TMS_RX_APS_TRANSACTIONS_1, and so on.
- To match the list of columns in UNION, select null values for the Pouch_Id, TC_Reference and Client_Reference columns that are only available in EPOSS Transaction tables.
- For Bureau Transactions of APPLICATION_TYPE = “BDC1”, the Quantity value needs to come from the “PURCHASED_QUANTITY” column, not the “QUANTITY” column.
- Outer join TPS Transaction and TPS_POL_FS_MAPPINGS tables on Prod_Id and Transaction_Mode_Id columns to determine the "Summarisation" flag value. TPS Transaction table will be the driving table in outer join.
- Use the latest mapping available in TPS_POL_FS_MAPPINGS table by checking that the Trading Date (at 08:00 hours) is between the Start Date and End Date (or End Date is null).
- Aggregate the transaction by Trading Date, Group Id, Product Id, Transaction Mode Id and Cash Account Period.
- Aggregate only those transactions where "Summarisation" flag for the Product and Mode is not set to "T".

The output of the SQL producing aggregated records will be merged (UNION ALL) with the transactions that are required unsummarised. The rules for producing Initial Summary for transactions required unsummarised are:

- Merge (UNION ALL) transactions from one partition of all TPS Transaction tables and select Trading_Date, Group_Id, Prod_Id, Transaction_Mode_Id, Cash Account Period, Quantity, Amount, Count of Transactions (equal to 1), Pouch_Id, TC_Reference and Client_Reference . For example, merge the transactions from TMS_RX_EPOSS_TRANSACTIONS_1, TMS_RX_APS_TRANSACTIONS_1, and so on.
- To match the list of columns in the UNION, select null values for the Pouch_Id, TC_Reference and Client_Reference columns that are only available in EPOSS Transaction tables.
- For Bureau Transactions of APPLICATION_TYPE = “BDC1”, the Quantity value needs to come from the “PURCHASED_QUANTITY” column, not the “QUANTITY” column.

FUJ00090946

- Join (simple join this time) TPS Transaction and TPS_POL_FS_MAPPINGS tables on Prod_Id and Transaction_Mode_Id columns to determine the "Summarisation" flag value. The Products and Modes, where Transaction data is required unsummarised, must have a corresponding mapping present in the TPS_POL_FS_MAPPINGS table.
- Use the latest mapping available in TPS_POL_FS_MAPPINGS table by checking that the Trading Date (at 08:00 hours) is between the Start Date and End Date (or End Date is null).
- Select only those transactions where "Summarisation" flag for the Product and Mode is set to "T".

For example, a part of SQL for Initial Summary may look like the following:

```
SELECT TXN.TRADING_DATE, TXN.GROUP_ID, TXN.PROD_ID, TXN.TRANSACTION_MODE_ID,
       TXN.CASH_ACCOUNT_PERIOD, SUM(TXN.QUANTITY), SUM(TXN.AMOUNT), COUNT(*),
       NULL /*POUCH_ID*/, NULL /*TC_REFERENCE*/, NULL /* CLIENT_REFERENCE*/
FROM (SELECT TRADING_DATE, GROUP_ID, PROD_ID, TRANSACTION_MODE_ID,
       CASH_ACCOUNT_PERIOD, QUANTITY, AMOUNT FROM TMS_RX_EPOSS_TRANSACTIONS_1
       UNION ALL
       SELECT TRADING_DATE, GROUP_ID, PROD_ID, TRANSACTION_MODE_ID,
       CASH_ACCOUNT_PERIOD, QUANTITY, AMOUNT FROM TMS_RX_APS_TRANSACTIONS_1
       UNION ALL
       .. .. ..
       .. .. ..
       SELECT TRADING_DATE, GROUP_ID, PROD_ID, TRANSACTION_MODE_ID,
       CASH_ACCOUNT_PERIOD, DECODE(APPLICATION_TYPE,
                                   'BDC1', PURCHASED_QUANTITY,
                                    QUANTITY),
       AMOUNT
       FROM TMS_RX_BDC_TRANSACTIONS_1) TXN,
TPS_POL_FS_MAPPINGS MAP
WHERE TXN.PROD_ID = MAP.PROD_ID (+)
       AND TXN.TRANSACTION_MODE_ID = MAP.TRANSACTION_MODE_ID (+)
       AND NVL(MAP.SUMMARISATION,'NULL') != 'T'
GROUP BY TXN.TRADING_DATE, TXN.GROUP_ID, TXN.PROD_ID,
       TXN.TRANSACTION_MODE_ID, TXN.CASH_ACCOUNT_PERIOD
UNION ALL
SELECT TXN.TRADING_DATE, TXN.GROUP_ID, TXN.PROD_ID, TXN.TRANSACTION_MODE_ID,
       TXN.QUANTITY, TXN.AMOUNT, 1 /*COUNT*/, TXN.CASH_ACCOUNT_PERIOD,
       TXN.POUCH_ID, TXN.TC_REFERENCE, TXN.CLIENT_REFERENCE
FROM (SELECT TRADING_DATE, GROUP_ID, PROD_ID, TRANSACTION_MODE_ID,
       CASH_ACCOUNT_PERIOD, QUANTITY, AMOUNT, POUCH_ID, TC_REFERENCE,
       CLIENT_REFERENCE FROM TMS_RX_EPOSS_TRANSACTIONS_1
       UNION ALL
       SELECT TRADING_DATE, GROUP_ID, PROD_ID, TRANSACTION_MODE_ID,
       CASH_ACCOUNT_PERIOD, QUANTITY, AMOUNT, NULL /*POUCH_ID*/,
       NULL /*TC_REFERENCE*/, NULL /* CLIENT_REFERENCE*/
       FROM TMS_RX_APS_TRANSACTIONS_1
       UNION ALL
       .. .. ..
       .. .. ..
       SELECT TRADING_DATE, GROUP_ID, PROD_ID, TRANSACTION_MODE_ID,
       CASH_ACCOUNT_PERIOD, DECODE(APPLICATION_TYPE,
                                   'BDC1', PURCHASED_QUANTITY,
                                   QUANTITY),
       AMOUNT, NULL /*POUCH_ID*/, NULL /*TC_REFERENCE*/,
       NULL /* CLIENT_REFERENCE*/
       FROM TMS_RX_BDC_TRANSACTIONS_1) TXN,
TPS_POL_FS_MAPPINGS MAP
WHERE TXN.PROD_ID = MAP.PROD_ID
AND TXN.TRANSACTION_MODE_ID = MAP.TRANSACTION_MODE_ID
AND MAP.SUMMARISATION = 'T'
```

A similar query will be executed for other partitions of the TPS Transaction tables. The data from the 1st partition of EPOSS, APS, OBCS, NWB, EFT and Bureau Transaction tables will be summarised into 1st partition of Initial Summary table. The

FUJ00090946

data from 2nd partition of Transaction tables will go to the 2nd partition of Initial Summary table, and so on. The data from 65th partition will be summarised into 65th partition of the Initial Summary table.

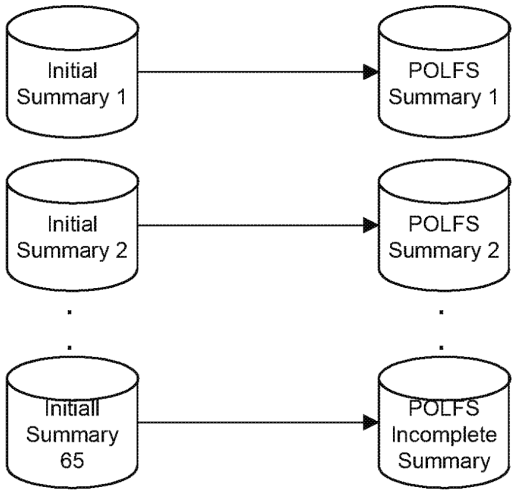
As required, the Cash Account Period will be only available during the migration period and therefore included in the summarisation. Once, all Branches have migrated to Monthly Branch Trading Statement and the Cash Account Period will not be available (set to null) and therefore will have no effect on the summarisation.

6.2.2 POL FS Summarisation Process (Final CBDB CAP onwards)

In this POL FS Summarisation, the Products and Transaction Modes in each partition of the Initial Summary tables will be mapped into POL FS Summaries as per the reference data in TPS_POL_FS_MAPPINGS table (refer to section 6.1.1).

Some Products and Modes where the transactions are required unsummarised (this attribute will be available via the "Summarisation" flag in the TPS_POL_FS_MAPPINGS table) will have the associated Account Reference Ids and/or Client Reference included in the summary.

Some transaction modes, for example, Transfer In and Transfer Out are not required by POL FS. All such modes will be present in the lookup table TPS_POL_FS_EXCLUDED_MODES table and excluded from the POL FS Summaries.



As a part of this summarisation, a number of associated attributes available via TPS_POL_FS_MAPPINGS will be included in the POL FS Summaries. Further, as defined by the reference data in TPS_POL_FS_MAPPINGS (MAP) table, data from multiple columns in TPS_PROD_MODE_SUMMARIES (PMS) table will be copied into a single column in TPS_POL_FS_SUMMARIES (PFS) table. For example, depending on the value of MAP.REFERENCE (PC = Pouch Bar Code, TC =

FUJ00090946

Transaction Correction Reference), either PMS.POUCH_ID or PMS.TC_REFERENCE will be copied into PFS.ACCOUNT_REFERENCE_ID.

The rules for generating POL FS Summaries are:

- Produce an inline select "SUM1" as
  - Simple join (one partition of) TPS_PROD_MODE_SUMMARIES (PMS) and TPS_POL_FS_MAPPINGS (MAP) tables on Prod_Id column to derive the value of Article_Id and other POL FS mapping attributes.
  - Select Trading_Date, Group_Id, Sum of Amount, Sum of Quantity[3] and Sum of Transaction Count from the TPS_PROD_MODE_SUMMARIES (TMS) table.
  - Select Article_Id from the TPS_POL_FS_MAPPINGS (MAP) table.
  - Select only those Initial Summary records where the corresponding "Summarisation" flag for the Product is set to 'S' in the TPS_POL_FS_MAPPINGS table.
  - Select only those Initial Summary records where the corresponding Cash_Account_Period is greater[4] than Final CBDB CAP or Cash Account Period is set to null in the TPS_PROD_MODE_SUMMARIES (PMS) table.
  - Select only those Initial Summaries where the Transaction Modes are not present in the lookup table TPS_POL_FS_EXCLUDED_MODES.
  - Aggregate the output by Trading Date, Group Id, Article Id.
- Join the output of SUM1 and TPS_POL_FS_MAPPINGS tables on Article_Id to derive other attributes required in the POL FS Summaries. Use a filter "Summarisation" flag equals 'S'. As this join may return more than one record with the same field values, select only one record using the DISTINCT clause.
- Select Trading_Date, Group_Id, Article_Id, Sum of Amount, Sum of Quantity and Sum of Transaction Count (used for SLA reporting) from the inline select "SUM1".
- Select Sub-file Id Flag as "BLCR1", as the CASH_ACCOUNT_PERIOD is greater than Final CBDB CAP.

---

[3] The 'QUANTITY' flag is an attribute of Article. So, all products belonging to an Article (and included in the sum) will have the same value of 'QUANTITY' flag, i.e. 'NA', 'CT', 'CQ' or 'QV'. Hence, the resultant sum will return a NULL (for 'NA'), Sum of Count (for 'CT'), Sum of Actual Quantity (for 'CQ') or Sum of Actual Amount **in Pence** (for 'QV'). The Sum of Quantity can be derived using the expression

```
SUM(DECODE(MAP.QUANTITY, 'NA', NULL,
                  'CT',PMS.TOTAL_TRANSACTION_COUNT,
                  'CQ', PMS.TOTAL_TRANSACTION_QUANTITY,
                  'QV', PMS.TOTAL_TRANSACTION_AMOUNT*100))
```

[4] NVL(PMS.CASH_ACCOUNT_PERIOD, 99) > *<FINAL CBDB CAP>*

FUJ00090946

- Select Article Type, Settlement Flag, Ledger Flag, Account, Movement_Type from the TPS_POL_FS_MAPPINGS table.
- To match the list of columns in the UNION, select null for Account_Reference_Id, Client_Reference, Pathway_Trans_Mode_Code_Type and Transaction_Mode_Id which are not available for POL FS Summaries at Article level.
- Use the latest mapping available in TPS_POL_FS_MAPPINGS table by checking that the Trading Date (at 08:00 hours) is between the Start Date and End Date (or End Date is null).

The output of the SQL producing aggregated records will be merged (UNION ALL) with the Summaries that are required at Article Mode level. The rules for producing these POL FS Summaries are:

- Produce an inline select "SUM2" as
    - Simple[5] join (one partition of) TPS_PROD_MODE_SUMMARIES (PMS) and TPS_POL_FS_MAPPINGS (MAP) tables on Prod_Id and Transaction_Mode_Id columns to derive the value of Article_Id and other POL FS mapping attributes.
    - Select Trading_Date, Group_Id, Transaction_Mode_Id, Sum of Amount, Sum of Quantity[6] and Sum of Transaction Count from the TPS_PROD_MODE_SUMMARIES (TMS) table.
    - Select Article_Id from the TPS_POL_FS_MAPPINGS (MAP) table.
    - Select only those Initial Summary records where the corresponding "Summarisation" flag for the Product is set to 'P' in the TPS_POL_FS_MAPPINGS table.
    - Select only those Initial Summary records where the corresponding Cash_Account_Period is greater[7] than Final CBDB CAP or Cash Account Period is set to null in the TPS_PROD_MODE_SUMMARIES (PMS) table.
    - Aggregate the output by Trading Date, Group Id, Article Id and Transaction_Mode_Id.

---

[5] See assumption in the section 3 - There is a mapping available for each Product and Transaction Mode where the transactions are required to be sent at Article Mode (Summarisation = 'P') or Unsummarised (Summarisation = 'T'). A check in RDDS ensures that this assumption is valid. Hence, the option of using outer join between TPS_PROD_MODE_SUMMARIES and TPS_POL_FS_MAPPINGS tables has been dropped.

[6] The 'QUANTITY' flag is an attribute of Article. So, all products belonging to an Article Mode (and included in the sum) will have the same value of 'QUANTITY' flag, i.e. 'NA', 'CT' 'CQ' or 'QV'. Hence, the resultant sum will return a NULL (for 'NA'), Sum of Count (for 'CT'), Sum of Actual Quantity (for 'CQ') or Sum of Actual Amount **in Pence** (for 'QV') .

[7] NVL(PMS.CASH_ACCOUNT_PERIOD, 99) > *<FINAL CBDB CAP>*

FUJ00090946

- Simple Join TPS_PROD_MODE_SUMMARIES and TPS_TRANS_MODE_CONVERSIONS table on Transaction_Mode_Id column to derive the value of Pathway_Transaction_Mode_Type_Code.
- Select Pathway_Transaction_Mode_Type_Code from the TPS_TRANS_MODE_CONVERSIONS table.
- Join the output of SUM2 and TPS_POL_FS_MAPPINGS tables on Article_Id and Transaction_Mode_Id to derive other attributes required in the POL FS Summaries. As this join may return more than one record with the same field values, select only one record using the DISTINCT clause
- Select Trading_Date, Group_Id, Article_Id, Transaction_Mode_Id, Sum of Amount, Sum of Quantity and Sum of Transaction Count (for SLA reporting) from the inline select "SUM2".
- Select Sub-file Id Flag as "BLCR1", as the CASH_ACCOUNT_PERIOD is greater than Final CBDB CAP.
- Select Article Type, Settlement Flag, Ledger Flag, Account, Movement_Type from the TPS_POL_FS_MAPPINGS table.
- To match the list of columns in the UNION, select null for Account_Reference_Id and Client_Reference which are not available for POL FS Summaries at Article Mode level.
- Use the latest mapping available in TPS_POL_FS_MAPPINGS table by checking that the Trading Date (at 08:00 hours) is between the Start Date and End Date (or End Date is null).
- Select only those Initial Summaries where the Transaction Modes are not present in the lookup table TPS_POL_FS_EXCLUDED_MODES.

The output of the SQL producing aggregated records will be merged (UNION ALL) with the Summaries that are required at Transaction level. The rules for producing these POL FS Summaries are:

- Simple join (one partition of) TPS_PROD_MODE_SUMMARIES and TPS_POL_FS_MAPPINGS tables on Prod_Id and Transaction_Mode_Id columns to derive various attributes required in POL FS Summaries.
- Simple Join TPS_PROD_MODE_SUMMARIES and TPS_TRANS_MODE_CONVERSIONS table on Transaction_Mode_Id column to derive the value of Pathway_Transaction_Mode_Type_Code.
- Select Trading_Date, Group_Id, Article_Id, Transaction_Mode_Id, Total Transaction Amount, Account_Reference_Id and Client_Reference from the TPS_PROD_MODE_SUMMARIES table.
- Select Pathway_Transaction_Mode_Type_Code from the TPS_TRANS_MODE_CONVERSIONS table.

FUJ00090946

- Select Sub-file Id Flag as "BLCR1", as the CASH_ACCOUNT_PERIOD is greater than Final CBDB CAP.
- Select Settlement Flag, Ledger Flag, Account, Movement_Type from the TPS_POL_FS_MAPPINGS table.
- Select Null, Total_Transaction_Count or Total_Transaction_Quantity from the TPS_PROD_MODE_SUMMARIES table as per the value of "Quantity" in the TPS_POL_FS_MAPPINGS table. For example,

```
DECODE(MAP.QUANTITY, 'NA', NULL,
            'CT', PMS.TOTAL_TRANSACTION_COUNT,
            'CQ', PMS.TOTAL_TRANSACTION_QUANTITY,
            'QV', PMS.TOTAL_TRANSACTION_AMOUNT*100)
```

- Select Sum of Transaction Count for SLA reporting.
- Select Null, Pouch_Id or TC_Reference from TPS_PROD_MODE_SUMMARIES table as per the value of "Reference" in the TPS_POL_FS_MAPPINGS table. For example,

```
DECODE(MAP.REFERENCE, 'NA', NULL,
            'PC', PMS.POUCH_ID,
            'TC', PMS.TC_REFERENCE)
```

- Use the latest mapping available in TPS_POL_FS_MAPPINGS table by checking that the Trading Date (at 08:00 hours) is between the Start Date and End Date (or End Date is null).
- Select only those Initial Summaries where "Summarisation" flag for the Product and Mode is set to "T" in the TPS_POL_FS_MAPPINGS.
- Select only those Initial Summary records where the corresponding Cash_Account_Period is greater[8] than Final CBDB CAP or Cash Account Period is set to null in the TPS_PROD_MODE_SUMMARIES (PMS) table.
- Select only those Initial Summaries where the Transaction Modes are not present in the lookup table TPS_POL_FS_EXCLUDED_MODES.

[8] NVL(PMS.CASH_ACCOUNT_PERIOD, 99) > *<FINAL CBDB CAP>*

FUJ00090946

For example, **a part of SQL** for POL FS Summary may look like the following:

```
SELECT DISTINCT SUM1.TRADING_DATE, SUM1.GROUP_ID, SUM1.ARTICLE_ID,
       SUM1.T_AMT, SUM1.T_QTY, SUM1.T_CNT, 'BLCR1' SUB_FILE_ID_FLAG,
       MAP.ARTICLE_TYPE, MAP.SETTLEMENT, MAP.LEDGER, MAP.ACCOUNT, MAP.MOVEMENT_TYPE,
       NULL /* ACCOUNT REFERENCE ID */, NULL /* CLIENT REFERENCE */,
       TO_NUM(NULL) /* TRANSACTION MODE ID */,
       NULL /* PATHWAY TRANS MODE TYPE CODE */
FROM   (SELECT PMS.TRADING_DATE, PMS.GROUP_ID, MAP.ARTICLE_ID,
               SUM(PMS.TOTAL_TRANSACTION_AMOUNT) T_AMT,
               SUM(DECODE(MAP.QUANTITY, 'NA', NULL,
                       'CT', PMS.TOTAL_TRANSACTION_COUNT,
                       'CQ', PMS.TOTAL_TRANSACTION_QUANTITY,
                       'QV', PMS.TOTAL_TRANSACTION_AMOUNT*100)) T_QTY,
               SUM(PMS.TOTAL_TRANSACTION_COUNT) T_CNT
        FROM   TPS_PROD_MODE_SUMMARIES_1 PMS, TPS_POL_FS_MAPPINGS MAP
        WHERE  PMS.PROD_ID = MAP.PROD_ID
        AND    MAP.SUMMARISATION = 'S'
        AND    NVL(PMS.CASH_ACCOUNT_PERIOD, 99) > <FINAL CBDB CAP>
        GROUP BY PMS.TRADING_DATE, PMS.GROUP_ID, MAP.ARTICLE_ID) SUM1,
        TPS_POL_FS_MAPPINGS MAP
WHERE  SUM1.ARTICLE_ID = MAP.ARTICLE_ID
AND    MAP.SUMMARISATION = 'S'
UNION ALL
SELECT DISTINCT SUM2.TRADING_DATE, SUM2.GROUP_ID, SUM2.ARTICLE_ID,
       SUM2.T_AMT, SUM2.T_QTY, SUM2.T_CNT, 'BLCR1' SUB_FILE_ID_FLAG,
       MAP.ARTICLE_TYPE, MAP.SETTLEMENT, MAP.LEDGER, MAP.ACCOUNT, MAP.MOVEMENT_TYPE,
       NULL /* ACCOUNT REFERENCE ID */, NULL /* CLIENT REFERENCE */,
       SUM2.TRANSACTION_MODE_ID, TMC.PATHWAY_TRANS_MODE_TYPE_CODE
FROM   (SELECT PMS.TRADING_DATE, PMS.GROUP_ID, MAP.ARTICLE_ID,
               PMS.TRANSACTION_MODE_ID,
               SUM(PMS.TOTAL_TRANSACTION_AMOUNT) T_AMT,
               SUM(DECODE(MAP.QUANTITY, 'NA', NULL,
                       'CT', PMS.TOTAL_TRANSACTION_COUNT,
                       'CQ', PMS.TOTAL_TRANSACTION_QUANTITY,
                       'QV', PMS.TOTAL_TRANSACTION_AMOUNT*100)) T_QTY,
               SUM(PMS.TOTAL_TRANSACTION_COUNT) T_CNT
        FROM   TPS_PROD_MODE_SUMMARIES_1 PMS, TPS_POL_FS_MAPPINGS MAP
        WHERE  PMS.PROD_ID = MAP.PROD_ID
        AND    PMS.TRANSACTION_MODE_ID = MAP.TRANSACTION_MODE_ID
        AND    MAP.SUMMARISATION = 'P'
        AND    NVL(PMS.CASH_ACCOUNT_PERIOD, 99) > <FINAL CBDB CAP>
        GROUP BY PMS.TRADING_DATE, PMS.GROUP_ID, MAP.ARTICLE_ID,
               PMS.TRANSACTION_MODE_ID) SUM2,
        TPS_POL_FS_MAPPINGS MAP,
        TPS_TRANS_MODE_CONVERSIONS TMC
WHERE  SUM2.ARTICLE_ID = MAP.ARTICLE_ID
AND    SUM2.TRANSACTION_MODE_ID = MAP.TRANSACTION_MODE_ID
AND    MAP.SUMMARISATION = 'P'
AND    SUM2.TRANSACTION_MODE_ID = TMC.POCL_TRANS_MODE_TYPE_CODE
UNION ALL
SELECT PMS.TRADING_DATE, PMS.GROUP_ID, MAP.ARTICLE_ID,
       PMS.TOTAL_TRANSACTION_AMOUNT,
       DECODE(MAP.QUANTITY, 'NA', NULL,
               'CT', PMS.TOTAL_TRANSACTION_COUNT,
               'CQ', PMS.TOTAL_TRANSACTION_QUANTITY,
               'QV', PMS.TOTAL_TRANSACTION_AMOUNT*100),
       PMS.TOTAL_TRANSACTION_COUNT,
       'BLCR1' SUB_FILE_ID_FLAG,
       MAP.ARTICLE_TYPE, MAP.SETTLEMENT, MAP.LEDGER, MAP.ACCOUNT, MAP.MOVEMENT_TYPE,
       DECODE(MAP.REFERENCE, 'NA', NULL,
               'PC', PMS.POUCH_ID,
               'TC', PMS.TC_REFERENCE),
       PMS.CLIENT_REFERENCE,
       PMS.TRANSACTION_MODE_ID, TMC.PATHWAY_TRANS_MODE_TYPE_CODE
FROM   TPS_PROD_MODE_SUMMARIES_1 PMS, TPS_POL_FS_MAPPINGS MAP,
       TPS_TRANS_MODE_CONVERSIONS TMC
WHERE  PMS.PROD_ID = MAP.PROD_ID
AND    PMS.TRANSACTION_MODE_ID = MAP.TRANSACTION_MODE_ID
AND    MAP.SUMMARISATION = 'T'
AND    NVL(PMS.CASH_ACCOUNT_PERIOD, 99) > <FINAL CBDB CAP>
AND    PMS.TRANSACTION_MODE_ID = TMC.POCL_TRANS_MODE_TYPE_CODE
```

The Group Id and Trading Date values above will be used to create the sub-files as required by POL FS AIS [R3]. The Sub-file Id Flag "BLCR1" indicates that the Cash

FUJ00090946

Account Period on the transaction is either greater than final CBDB CAP or Null (as CAP will not be available in future).

Cash Account Period has been used to decide whether an initial summary record needs to be included in this POL FS feed, as only cash and near cash products need to be sent until the final CBDB CAP, say 'N'. During the migration period when a mixture of transactions for CAP <= 'N' or CAP>'N' are received, this process will set the Sub-file Id Flag in POL_FS_SUMMARIES table to "BLCR1". Initial Summaries or CAP <= 'N' will be processed via a separate process (see 6.2.3 for details).

There will be multiple instances of the POL FS summarisation process running in parallel, viz., eight instances each processing eight table partitions.

The data from the 1st partition of Initial Summary table will be summarised into 1st partition of POL FS Summary table. The data from 2nd partition of Initial Summary table will go to the 2nd partition of POL FS Summary table, and so on. The data from 65th partition of Initial Summary table will be summarised into POL FS Incomplete Summary table.

6.2.3 POL FS Summarisation Process (Before Final CBDB CAP)

This process will generate POL FS Summaries for Cash and Near Cash Products where CAP <= "Final CBDB CAP" using the locally defined mappings available in TPS_POL_FS_MAPPINGS_AT_S60 (refer to section 6.1.1).

The process will extract data from the Initial Summary tables which have aggregated data based on S80 mappings in TPS_POL_FS_MAPPINGS table. **Therefore, the process assumes that there are no products (and Transaction Modes) which were required unsummarised at S60 but are summarised at S80.**

The rules for generating POL FS Summaries for CAP <= "Final CBDB CAP" will be same as those for POL FS Summarisation process (TPSC291) for CAP > "Final CBDB CAP except the following:

- Use S60 POL FS mappings defined in TPS_POL_FS_MAPPINGS_AT_S60 table in place of TPS_POL_FS_MAPPINGS table.
- Select Sub-file Id Flag as "BLCR3", as the CASH_ACCOUNT_PERIOD is less than or equal to Final CBDB CAP.
- Select only those Initial Summary records where the corresponding Cash_Account_Period is less than or equal to the Final CBDB CAP.
- For the summaries that are required at Article Mode or Transaction level, **simple join** the TPS_PROD_MODE_SUMMARIES and TPS_POL_FS_MAPPINGS_AT_S60 tables on Prod_Id and Transaction_Mode_Id columns, as all products required in S60 summaries must be present in the TPS_POL_FS_MAPPINGS_AT_S60 table.
- In this case all entries will have summarisation of “S” or “T”. So, the additional complexities of Summarisation = “P” can be ignored

FUJ00090946

For example, **a part of SQL** for POL FS Summary may look like the following:

```
SELECT DISTINCT SUM1.TRADING_DATE, SUM1.GROUP_ID, SUM1.ARTICLE_ID,
       SUM1.T_AMT, SUM1.T_QTY, SUM1.T_CNT, 'BLCR3' SUB_FILE_ID_FLAG,
       MAP.ARTICLE_TYPE, MAP.SETTLEMENT, MAP.LEDGER, MAP.ACCOUNT, MAP.MOVEMENT_TYPE,
       NULL /* ACCOUNT REFERENCE ID */, NULL /* CLIENT REFERENCE */,
       0 /* TRANSACTION MODE ID */, NULL /* PATHWAY TRANS MODE TYPE CODE */
FROM   (SELECT PMS.TRADING_DATE, PMS.GROUP_ID, MAP.ARTICLE_ID,
               SUM(PMS.TOTAL_TRANSACTION_AMOUNT) T_AMT,
               SUM(DECODE(MAP.QUANTITY, 'NA', 0,
                       'CT', PMS.TOTAL_TRANSACTION_COUNT,
                       'CQ', PMS.TOTAL_TRANSACTION_QUANTITY,
                       'QV', PMS.TOTAL_TRANSACTION_AMOUNT*100)) T_QTY,
               SUM(PMS.TOTAL_TRANSACTION_COUNT) T_CNT
        FROM   TPS_PROD_MODE_SUMMARIES_1 PMS, TPS_POL_FS_MAPPINGS_AT_S60 MAP
        WHERE  PMS.PROD_ID = MAP.PROD_ID
        AND    MAP.SUMMARISATION = 'S'
        AND    PMS.CASH_ACCOUNT_PERIOD <= <FINAL CBDB CAP>
        GROUP BY PMS.TRADING_DATE, PMS.GROUP_ID, MAP.ARTICLE_ID) SUM1,
        TPS_POL_FS_MAPPINGS_AT_S60 MAP
WHERE  SUM1.ARTICLE_ID = MAP.ARTICLE_ID
AND    MAP.SUMMARISATION = 'S'
UNION ALL
SELECT PMS.TRADING_DATE, PMS.GROUP_ID, MAP.ARTICLE_ID,
       PMS.TOTAL_TRANSACTION_AMOUNT,
       DECODE(MAP.QUANTITY, 'NA', 0,
               'CT', PMS.TOTAL_TRANSACTION_COUNT,
               'CQ', PMS.TOTAL_TRANSACTION_QUANTITY,
               'QV', PMS.TOTAL_TRANSACTION_AMOUNT*100),
       PMS.TOTAL_TRANSACTION_COUNT,
       'BLCR1' SUB_FILE_ID_FLAG,
       MAP.ARTICLE_TYPE, MAP.SETTLEMENT, MAP.LEDGER, MAP.ACCOUNT, MAP.MOVEMENT_TYPE,
       DECODE(MAP.REFERENCE, 'NA', NULL,
               'PC', PMS.POUCH_ID,
               'TC', PMS.TC_REFERENCE),
       PMS.CLIENT_REFERENCE,
       PMS.TRANSACTION_MODE_ID, TMC.PATHWAY_TRANS_MODE_TYPE_CODE
FROM   TPS_PROD_MODE_SUMMARIES_1 PMS, TPS_POL_FS_MAPPINGS_AT_S60 MAP,
       TPS_TRANS_MODE_CONVERSIONS TMC
WHERE  PMS.PROD_ID = MAP.PROD_ID
AND    PMS.TRANSACTION_MODE_ID = MAP.TRANSACTION_MODE_ID
AND    MAP.SUMMARISATION = 'T'
AND    PMS.CASH_ACCOUNT_PERIOD <= <FINAL CBDB CAP>
AND    PMS.TRANSACTION_MODE_ID = TMC.POCL_TRANS_MODE_TYPE_CODE
```

The Group Id and Trading Date values above will be used to create the sub-files as required by POL FS AIS [R3]. The Sub-file Id Flag "BLCR3" indicates that the Cash Account Period on the transaction is less than or equal to the final CBDB CAP. As a part of the file generation process, a Balancing Transaction will be generated to the account '999999' such that the sub-file totals balance to zero.

There will be multiple instances of this POL FS summarisation process running in parallel, viz., eight instances each processing eight table partitions.

The data from the 1st partition of Initial Summary table will be summarised into 1st partition of POL FS Summary table. The data from 2nd partition of Initial Summary table will go to the 2nd partition of POL FS Summary table, and so on. The data from 65th partition of Initial Summary table will be summarised into POL FS Incomplete Summary table.

Having a separate process (as opposed to merging this functionality into TPSC291) will help in implementing any special rules which may be needed for the pre-migration POL FS Summaries. Also, this process can be turned off cleanly after the migration.

FUJ00090946

6.2.4 Produce Opening Figures for POL FS Summaries (at Final CBDB CAP)

This process will generate Opening Figures for POL FS Summaries at Migration Point 30 (see section 6.8.4). The Opening Figures will be derived from the Cash Account and Stock Holding records for CAP = "Final CBDB CAP" as per the reference data in TPS_POL_FS_MAPPINGS table (refer to section 6.1.1).

The corresponding summary records will have the Sub-file Id Flag set to "BLCR2".

The Cash Account information harvested into TPS is split into two parts:

- Stock Holding information (STX records in TMS_RX_STOCK_HOLDINGS tables)
- Cash Account information (CAC records in TMS_RX_CASH_ACCOUNTS tables)

POL FS requires opening figures for all Stock products, plus the initial position for all Suspense and Discrepancy products.

Data for the Stock products can be obtained from the STX records, while details of the Suspense and Discrepancy products can be obtained from the CAC records where line numbers are 50xx. No opening figures are required for [CP3884 requires opening figures for Cash and Cash in Pouches to be sent] Cheques since POL FS will already have this data from Impact R1.

PC115574 – As the products Cash (1) Cash-in-Pouches are selected from the Stock Holdings table and Cash Accounts table respectively, but map to the same Article, the SQL (in section 6.2.4.2) needs a further level of summarisation, i.e., by Branch, Trading Date and Article Id.

PC115380 – Products 145 (Surplus Discrepancy) and 222 (Loss System) are harvested via both Stock Holdings and Cash Accounts route. Therefore, line numbers 07xx are not required for generating Opening Figures. The related Opening Figures can be generated from the associated Stock Holding lines for these products.

6.2.4.1 Opening figures from Stock Holding Records

As the STX records have no concept of Mode, it will set to (as suggested and agreed) RIAD (24) or Default (0)[9] in the SQL. Further, there should be no mappings used where Summarisation = T. This means that the Summarisation flag in TPS_POL_FS_MAPPINGS table can be ignored while generating the Opening Figures from STX records.

POL FS requires new opening figures for all Foreign Exchange (Bureau) products. This is because a single total was maintained for all currencies at S60 but separate per-currency figures are required at S80. These opening figures will be based on a new

[9] If a product has a default mapping (Transaction Mode = 0), no mappings for specific Transaction Modes can exist for the Product. See section 3.

FUJ00090946

'Purchased Quantity' column added to the TMS_RX_STOCK_HOLDING tables. TPS Harvester will change to populate this column with <AdditionalData.BDC.PQty:> attribute from the Opening Figures message.

The rules for generating Opening Figures from Stock Holding records will be:

- Produce an inline select "SUM1" as
    - Simple join (one partition of) TMS_RX_STOCK_HOLDINGS (TSH) and TPS_POL_FS_MAPPINGS (MAP) tables on Prod_Id column to derive the value of Article_Id and any other POL FS mapping attributes.
    - Put a filter on MAP.TRANSACTION_MODE_ID to be either 24 (RIAD) or 0 (Default).
    - If MAP.QUANTITY flag is set to 'CT', then select 1 as the Sum of Quantity. [PC115465] If MAP.QUANTITY flag is set to 'QV', then select Amount in Pence as the Sum of Quantity. Otherwise, select Sum of Purchased Quantity (if this column contains a value) or Sum of Quantity from the TMS_RX_STOCK_HOLDINGS (TSH) table.
    - Select Trading_Date, Group_Id, Sum of Amount from the TMS_RX_STOCK_HOLDINGS (TSH) table.
    - Select Article_Id from the TPS_POL_FS_MAPPINGS (MAP) table.
    - Select only those Stock Holding records where the TSH.STOCK_TYPE is **NOT** 'L' (low-level denominations).
    - Select only those Stock Holding records where the TSH.PROD_ID is **NOT** 2 (Cheques).

    [DN: CP3884 - Opening Figures for Cash and Cash in Pouches are required.]

    - Do **NOT** select those Stock Holding records where Purchased Quantity is NULL and Quantity is NULL or zero (this has been discussed and agreed with the Xansa).
    - Select only those Stock Holding records where the TSH.CASH_ACCOUNT_PERIOD is equal to Final CBDB CAP.
    - Aggregate the output by Trading Date, Group Id and Article Id.
- Join the output of SUM1 and TPS_POL_FS_MAPPINGS tables on Article_Id to derive other attributes required in the POL FS Summaries.
- Select Trading_Date, Group_Id, Article_Id, Sum of Amount, Sum of Quantity and 1 (for Sum of Transaction Count) from the inline select "SUM1".
- Select Sub-file Id Flag as "BLCR2", as the CASH_ACCOUNT_PERIOD is equal to Final CBDB CAP.
- Select Article Type, Settlement Flag, Ledger Flag, Account, Movement_Type from the TPS_POL_FS_MAPPINGS table.

FUJ00090946

- Use the latest mapping available in TPS_POL_FS_MAPPINGS table by checking that the Trading Date (at 08:00 hours) is between the Start Date and End Date (or End Date is null).

6.2.4.2 Opening figures from Cash Account Records

As the Product Ids are not available in Cash Account (CAC) records harvested into TPS, a locally defined mapping[10] (see the spreadsheet in the Appendix F) will be required from these line numbers to the appropriate products to be used. Once the Product Id has been obtained, the data can be summarised onto Articles using the mappings defined in TPS_POL_FS_MAPPINGS table.

As the CAC records have no concept of Mode, it will set (as suggested and agreed) to HK (15) or Default (0)[11] in the SQL. Further, there should be no mappings used where Summarisation = T. This means that the Summarisation flag in TPS_POL_FS_MAPPINGS table can be ignored while generating the Opening Figures from CAC records.

The rules for generating Opening Figures from Cash Account records will be:

- Produce an inline select "SUM1" as:
    - Simple join (one partition of) TMS_RX_CASH_ACCOUNTS (TCA) and TPS_OPENFIG_CAC_PROD_MAPPINGS (TCP) tables on the LINE_NUMBER column to derive the value of TCP.PROD_ID. The simple join will also act as a filter and select only those Cash Account records where the corresponding LINE_NUMBER is present[12] in the TPS_OPENFIG_CAC_PROD_MAPPINGS table.
    - Simple join (one partition of) TMS_RX_CASH_ACCOUNTS (TCA) and TPS_POL_FS_MAPPINGS (MAP) tables on PROD_ID column to derive the value of Article_Id and any other POL FS mapping attributes.
    - Put a filter on MAP.TRANSACTION_MODE_ID to be either 15 (HK) or 0 (Default).

---

[10] This could be obtained from Reference Data, however the mapping of Product to Line number is many to one, so it may be better to explicitly define a product to use for each of the possible line numbers. There is a maximum of 200 such lines, though in practice there are only about 30 defined as being used, and hence needing to have a mapping defined for them.

[11] I have confirmed from RDDS Design that if a product has a default mapping in Mode 0, i.e. if summarisation is required at the product level (Summarisation='S'), this will be the only mapping available in the POL FS Mappings table/view. For example, if the product 2846 has a default mapping in Mode 0, it can not have another mapping in Mode 24 or 15, and vice versa. This means mappings in Mode 0 and any other Mode (15, 24, etc.) are mutually exclusive. Hence, a filter on Modes (0, 15) for Suspense product and (0, 24) for Stock products will return only one mapping from the POL FS Mappings table. See section 3.

[12] These Line Numbers would be between 700 and 799 or 5000 and 5099. The individual Line Numbers will be present in TPS_OPENFIG_CAC_PROD_MAPPINGS table to avoid hard coding in the program.

FUJ00090946

- Multiply the Cash Account Line Amount by 1 if TCP.ACCOUNTING_SENSE is "+1" or -1 if accounting sense is "-1". Select Trading_Date, Group_Id, Sum of Amount from the TMS_RX_CASH_ACCOUNTS (TCA) table.
- Select Article_Id from the TPS_POL_FS_MAPPINGS (MAP) table.
- Select only those Cash Account records where the TCA.CASH_ACCOUNT_PERIOD is equal to Final CBDB CAP.
- Aggregate the output by Trading Date, Group Id and Article Id.

- Join the output of SUM1 and TPS_POL_FS_MAPPINGS tables on Article_Id to derive other attributes required in the POL FS Summaries.
- Select Trading_Date, Group_Id, Article_Id, Sum of Amount and 1 (for Sum of Transaction Count) from the inline select "SUM1".
- If MAP.QUANTITY flag is set to 'NA', then select NULL as the Sum of Quantity. [PC115465] If MAP.QUANTITY flag is set to 'QV', then select Amount in Pence as the Sum of Quantity. Otherwise, select 1 as the Sum of Quantity.
- Select Sub-file Id Flag as "BLCR2", as the CASH_ACCOUNT_PERIOD is equal to Final CBDB CAP.
- Select Article Type, Settlement Flag, Ledger Flag, Account, Movement_Type from the TPS_POL_FS_MAPPINGS table.
- Use the latest mapping available in TPS_POL_FS_MAPPINGS table by checking that the Trading Date (at 08:00 hours) is between the Start Date and End Date (or End Date is null).

FUJ00090946

For example, **a part of SQL** for POL FS Summary may look like the following:

```
SELECT DISTINCT SUM1.TRADING_DATE, SUM1.GROUP_ID, SUM1.ARTICLE_ID,
       SUM1.T_AMT, SUM1.T_QTY, 1 T_CNT, 'BLCR2' SUB_FILE_ID_FLAG,
       MAP.ARTICLE_TYPE, MAP.SETTLEMENT, MAP.LEDGER, MAP.ACCOUNT, MAP.MOVEMENT_TYPE,
       NULL /* ACCOUNT REFERENCE ID */, NULL /* CLIENT REFERENCE */,
       0 /* TRANSACTION MODE ID */, NULL /* PATHWAY TRANS MODE TYPE CODE */
FROM   (SELECT TSH.TRADING_DATE, TSH.GROUP_ID, MAP.ARTICLE_ID,
               SUM(TSH.AMOUNT) T_AMT,
               SUM(DECODE(MAP.QUANTITY, 'CT', 1,
                          MAP.QUANTITY, 'QV', TSH.AMOUNT*100,
                          NVL(TSH.PURCHASED_QUANTITY, TSH.QUANTITY))) T_QTY
        FROM   TMS_RX_STOCK_HOLDINGS_1 TSH, TPS_POL_FS_MAPPINGS MAP
        WHERE  TSH.PROD_ID = MAP.PROD_ID
        AND    MAP.TRANSACTION_MODE_ID IN (0, 24) /* DEFAULT OR RIAD */
        AND    TSH.CASH_ACCOUNT_PERIOD = <FINAL CBDB CAP>
        AND    TSH.STOCK_TYPE != 'L'
        AND    TSH.PROD_ID = 2 /* CHEQUE */
        AND    (TSH.PURCHASED_QUANTITY IS NOT NULL AND NVL(QUANTITY,0) != 0)
        GROUP BY TSH.TRADING_DATE, TSH.GROUP_ID, MAP.ARTICLE_ID) SUM1,
        TPS_POL_FS_MAPPINGS MAP
WHERE  SUM1.ARTICLE_ID = MAP.ARTICLE_ID
AND    MAP.TRANSACTION_MODE_ID IN (0,24) /* DEFAULT OR RIAD */
UNION ALL
SELECT DISTINCT SUM1.TRADING_DATE, SUM1.GROUP_ID, SUM1.ARTICLE_ID,
       SUM1.T_AMT, SUM1.T_QTY, 1 T_CNT, 'BLCR2' SUB_FILE_ID_FLAG,
       MAP.ARTICLE_TYPE, MAP.SETTLEMENT, MAP.LEDGER, MAP.ACCOUNT, MAP.MOVEMENT_TYPE,
       NULL /* ACCOUNT REFERENCE ID */, NULL /* CLIENT REFERENCE */,
       0 /* TRANSACTION MODE ID */, NULL /* PATHWAY TRANS MODE TYPE CODE */
FROM   (SELECT TCA.TRADING_DATE, TCA.GROUP_ID, MAP.ARTICLE_ID,
               SUM(TCA.AMOUNT*TCP.ACCOUNTING_SENSE) T_AMT,
               SUM(DECODE(MAP.QUANTITY, 'NA', TO_NUMBER(NULL),
                          MAP.QUANTITY, 'QV', TCA.AMOUNT*TCP.ACCOUNTING_SENSE*100,
                          1)) T_QTY
        FROM   TMS_RX_CASH_ACCOUNTS_1 TCA, TPS_POL_FS_MAPPINGS MAP,
               TPS_OPENFIG_CAC_PROD_MAPPINGS TCP
        WHERE  TCA.LINE_NUMBER = TCP.LINE_NUMBER
        AND    TCP.PROD_ID = MAP.PROD_ID
        AND    MAP.TRANSACTION_MODE_ID IN (0, 15) /* DEFAULT OR HK */
        AND    TCA.CASH_ACCOUNT_PERIOD = <FINAL CBDB CAP>
        GROUP BY TCA.TRADING_DATE, TCA.GROUP_ID, MAP.ARTICLE_ID) SUM1,
        TPS_POL_FS_MAPPINGS MAP
WHERE  SUM1.ARTICLE_ID = MAP.ARTICLE_ID
AND    MAP.TRANSACTION_MODE_ID IN (0, 15) /* DEFAULT OR HK */
```

There will be multiple instances of this POL FS summarisation process running in parallel, viz., eight instances each processing eight table partitions.

The data from the 1st partition of Stock Holdings/Cash Accounts tables will be summarised into 1st partition of POL FS Summary table. The data from 2nd partition of Stock Holdings/Cash Accounts tables will go to the 2nd partition of POL FS Summary table, and so on.

The Group Id and Trading Date values above will be used to create the sub-files for Opening Figures. The Opening Figures will be kept separate from normal transactions and will be passed over in a sub-file with a different Header type BLCR2 as defined in the POL FS AIS [R3]. Two Balancing Transactions will be generated as detailed in the section 6.3.3 such that the sub-file totals balance to zero.

For simplicity, these sub-files will be sent to POL FS in a separate data file. The file and record formats will be as per POL FS AIS [R3]. The details of this file will be included in the EOT file for the transmission.

As the Branch Cash Accounts 'N' may be received over a number of days, the Opening Figures may be sent to POL FS over a number of days.

FUJ00090946

POL will be required to ensure that all Branches generate the final Cash Account on time. This will ensure that we have a clean cutover.

The final (CBDB) Cash Account will be stored as a TPS System Parameter (configurable). During the migration period, the TPS_TIP_SUB_FILE_REGISTER can be used to identify which outlets have produced the final Cash Account.

Once all branches have produced the Final CBDB CAP, i.e., no CAC or STX records are expected for a CAP = "Final CBDB CAP", this program can be switched off (this can be done at migration point 40).

6.3 Producing the POL FS Summary Data and EOT Files

The POL FS Summaries will be sent in flat data files. The data will be read from POL FS Summary table and written to files as per the record formats present in POL FS AIS [R3]. A data file will contain one or more sub-files. There will be one sub-file per Branch and Trading Day.

Zero value summary records (refer to the section "Suppressing Zero Value Records" in [R3] for the rules) will not be included the sub-files. Further, if **all** summary records for a sub-file are zero value (i.e., there are none to include in the sub-file), then the sub-file header (i.e., BLE Content Record) for this sub-file will also be not written. Empty files (containing File Header and Trailer only, no sub-files) will be deleted and not sent to POL FS.

On a business day, there will be up to 64 data files from the normal POL FS Summary tables. Also, if there are incomplete summaries from a previous day, which have now been corrected, then a 65th data file will be generated from the POL FS Incomplete Summary table. It will be better to have a separate file for the repaired/corrected POL FS Summaries than merging them in one of the normal POL FS Summary files for the day. During the migration, there will be additional data files containing "Pre-migration Summaries" and "Opening Figures" (see section 6.8). All data files will be created in a predefined host directory.

At the end of each day's transmission an EOT (End of Transmission) file will be created and sent. The EOT file will state the total count/names of data files in the transmission (the count/names do not include the end of transmission file). This file will be the last in the transmission and will act as a final "handshake" between the two systems. The file and record formats and complete set of validation rules are present in POL FS AIS [R3].

FUJ00090946

6.3.1 Producing the Data File from POL FS Incomplete Summaries Table

Before processing any normal POL FS Summary and producing POL FS data files for the day, the POL FS Incomplete Summary table will be checked to see if it contains any Branch and Trading Date where there are no pending harvester exceptions and the net value of POL FS Summaries now balances to zero (for BLCR1 sub-file type only). All such summary records will be written to a separate file (65th file) and included in the day's transmission.

The advantage of processing Incomplete Summaries first is that any new records inserted today into POL FS Incomplete Summaries table will not be revisited.

For the post-migration summaries (Sub-file Id flag BLCR1):

- Check if there is a matching record for the Group_Id and Trading_Date where Record Repaired flag is set to 'N' in the TMS_HARVESTER_EXCEPTIONS_ARC table.
- Check if the net value of summaries for the Group_Id and Trading_Date still does not balance to zero.

If any of the above conditions is true, the summaries will be held back.

For the pre-migration summaries and opening figures (Sub-file Id flags BLCR3 and BLCR2 respectively):

- Check if there is a matching record for the Group_Id and Trading_Date where Record Repaired flag is set to 'N' in the TMS_HARVESTER_EXCEPTIONS_ARC table.

If the above condition is true, the summaries will be held back.

For all Outlets and Trading Dates where there are no pending harvester exceptions (for BLCR1, BLCR2 or BLCR3 sub-file types) and the net value of POL FS Summaries now balances to zero (only for BLCR1 sub-file type):

- all summary records which are sent summarised and have the same key values (Article Id and Transaction Mode Id) will be aggregated so that only one record per key value is written to the data file.
- a sub-file containing all summary records for the Branch and Trading Date will be created in the data file.
- For the pre-migration summaries and opening figures (Sub-file Id flags BLCR3 and BLCR2), appropriate balancing transactions will be generated as detailed in the section 6.3.3.
- an entry will be made in the POL FS Sub-file Register for the Branch and Trading Date. The record will have TOTAL_RECORDS column set to the number of summary records and TOTAL_TRANSACTION_COUNT column set to the sum of TOTAL_TRANSACTION_COUNT for the summary records in the sub-file.

FUJ00090946

- all summary records written to the data file will be logically deleted from the POL FS Incomplete Summary table by setting the "PROCESSED" flag to 'Y' for each record. These records will be physically deleted from the table at TPS End of Day.

6.3.2 Producing the Data Files from Normal POL FS Summary Tables

Multiple processes will be run in parallel to read data from the normal POL FS Summary table and write to files as per the record formats present in POL FS AIS [R3]. A data file will contain one or more sub-files. There will be one sub-file per Branch and Trading Day. The data files will be created in a predefined host directory. There will be up to 64 data files (one per normal table partition) from the normal POL FS Summary tables.

As mentioned in POL FS AIS [R3], the sub-file totals within a data file must balance to zero for post-migration summaries (Sub-file Id flag BLCR1). If this is not the case, the transactions are known to be missing, so there is no point in attempting to produce a sub-file for passing to POL FS since it will be incomplete. One of the possible reasons for a sub-file being incomplete (sub-file total not balancing to zero) is that some transactions have failed Harvesting due to a database check constraint failure. In such cases, the transactions will have to be repaired before they are passed through. This means, these transactions will not be available immediately for summarisation.

The post-migration (Sub-file Id flag BLCR1) POL FS Summaries, where the net value for a Branch and Trading Date does not balance to zero, will be held back for correction. Such cases will be detected by first examining the TPS Harvester Exception tables and then summing up the summary values for the Branch and Trading Date. If there is a pending harvester exception or the summary total does not balance to zero then all summary records for that Branch and Trading Date will be inserted into POL FS Incomplete Summary table.

Also, the pre-migration (Sub-file Id flag BLCR3) POL FS Summaries and Opening Figures (Sub-file Id flag BLCR2), where a harvester exception exits, will need to be held back until the harvester exception is repaired and included in the summary. This is because POL FS should ever receive only one sub-file for a Group Id and Trading Date. Such cases will be detected by examining the TPS Harvester Exception tables for the Branch and Trading Date. If there is a pending harvester exception then all summary records for that Branch and Trading Date will be inserted into POL FS Incomplete Summary table.

To check if there is a pending harvester exception, the overall view[13] on TMS_HARVESTER_EXCEPTIONS (partitions 1 to 64) tables will be searched for the Group_Id and Trading_Date. At the end of process, an operational exception will be raised showing the number of incomplete sub-files found during the run.

[13] PC113637 – Agent users' mappings for Transaction and Harvester Exception table partitions are different. Hence, the exceptions may not be in the same partition number as the transactions.

FUJ00090946

At present, only the Trading Date is explicitly stored in the Harvester Exceptions table. The Group Id, Amount and Quantity are stored in the generic (COLUMN_NAME_n, COLUMN_VALUE_n) columns. To implement the above check, the Group Id will need to be stored explicitly in the Harvester Exceptions table. Also, an explicit flag[14] was required to indicate that the harvester exception has a financial value, i.e., has a non-zero Quantity and/or Amount. But, it is a bit harder for the TPS Harvester to populate the Financial Transaction flag in any exceptions it may raise. Given that a very few harvester exceptions are expected after relaxing the check constraints on CAP and BP and out of these exceptions raised, 99% would have this flag to be set to 'Y', it has been decided to remove this flag altogether. POL FS Summaries for a Branch and Trading Date will be held back if there are any harvester exceptions for these.

TPS Harvester will need to change to populate the Group Id into explicit columns in TMS_HARVESTER_EXCEPTIONS table.

A new column GROUP_ID will be added to TMS_HARVESTER_EXCEPTIONS_nn (nn=1 to 64) and TMS_HARVESTER_EXCEPTIONS_ARC tables.

If the post-migration POL FS Summaries for a Branch and Trading Date balance to zero, then they will be written to a sub-file in the data file. As described in POL FS AIS [R3], the definition of Negative and Positive Accounting Sense in the Horizon system is reverse to the accounting required by the POL Ledgers. Therefore, the sign on the Label Identifier (BLDR- and BLDR+) in the summary records will be the reverse of normal Horizon signs. Also, the amounts will be converted to pence from pounds and pence before writing to the summary records.

A record will be created in the POL FS Sub-file Register table for each Branch and Trading Date processed by the program. The record will have TOTAL_RECORDS column set to the number of summary records and TOTAL_TRANSACTION_COUNT column set to the sum of TOTAL_TRANSACTION_COUNT for the summary records in the sub-file.

For the pre-migration summaries and opening figures (Sub-file Id flags BLCR3 and BLCR2), appropriate balancing transactions will be generated as detailed in the section 6.3.3.

6.3.3 Producing the Balancing Transaction for Pre-migration Summaries (BLCR3) and Opening Figures (BLCR2)

The pre-migration summaries (Sub-file Id flag BLCR3) will require an appropriate "Balancing Transaction" to account 999999 such that the sub-files sent to POL FS

[14] It is a bit harder for the TPS Harvester to set the Financial Transaction Flag in any exception it may raise. Given that I expect 99% of all transactions to require this flag to be set and also that I would expect very few exceptions, I would rather just remove this flag altogether and for TPS Host to assume that any exception has a 'financial effect'.

FUJ00090946

balances to zero. For example, if the Account Value for all detail records in a sub-file is equal to -203.88, a Balancing Transaction with the Account Value of +203.88 will be generated.

For the Opening Balances (Sub-file Id flag BLCR2), which contain data for both suspense and stock products, two separate Balancing Transaction are required - one for the suspense products and one for the stock products. These Balancing Transactions will be generated as follows:

- Within the sub-file (for a Branch and Trading Date), divide the summary records into two groups - one where the "Account" column has a NULL value and one where the "Account" column has a value.
- For the Group where the "Account" column has a NULL value, generate a Balancing Transaction to account 999999.
- For the Group where the "Account" column has a value, generate a Balancing Transaction to account 999998.

The details (like Settlement, Ledger, etc.) of the Balancing Transaction will be stored in TPS_POL_FS_MAPPINGS_AT_S60 table as per contents of the spreadsheet in the Appendix E.

6.3.4 Producing the EOT File

Once all the POL FS Summary data files have been created (including any opening balance file described later), a separate program (similar to that at S60) will create and End of Transmission File. The EOT file will contain the names of data files in the transmission (the names do not include the end of transmission file). The file and record formats are present in POL FS AIS [R3].

This program will also create:

- a link for the file in the audit directory pointed to by environment variable POLFS_AUDIT.
- a copy of the file in the POL FS directory NFS mounted on the host (hard links may not span file systems). This directory will be pointed to by environment variable POLFS_SHARE.

Once a file is successfully copied to POLFS_SHARE directory, the program will record date and time of the file copy into TPS File Register. This date and time will be used in file delivery information generated for Data Warehouse (Fujitsu Service).

The copy in POLFS_SHARE directory will only be created if there are no EOT files present in the POL FS directory. That is, the program will check for the presence of an EOT file in the POL FS directory before creating any copies of the Summary files or EOT file for the day. If an EOT file is found in the POLFS_SHARE directory, the program will raise an operational exception to alert SSC and continue creating the files in the host directory. The program will exit a success even if it could not create copies in the POLFS_SHARE directory so that the TPS schedule is not held up.

FUJ00090946

As a result of the operational exception, SSC will check the contents of POLFS_SHARE directory, get the pending files processed by POL FS and then manually copy all Summary files and associated EOT file for the day, present in the POLFS_OUTPUT directory.

6.3.5 Host Directories and File Names for POL FS Summary and EOT Files

The naming convention for the POL FS Summary and EOT files is present in POL FS AIS [R3], but repeated here for convenience.

File Type	File Name
POL FS Summary	IFccyymmdd999.BLE
EOT File	IFccyymmddnnn.EOT

Where,
ccyymmdd: Represents the TPS Business Day (derived from the TPS System Parameter "TPS SYSTEM DATE")

999: Represents the BLE Data File Sequence Number (derived from the TPS System Parameter "POL FS BLE FILE SEQUENCE". The file sequence will range from 001-064 for normal data files on a TPS business day, 065 for previously incomplete summaries, and a different number series[15] for manually repaired data files; reset to 001 by TPS End of Day process).

nnn: Represents the EOT Data File Sequence Number (derived from the TPS System Parameter "POL FS EOT FILE SEQUENCE"; set to 000 for the EOT file for each TPS business day; reset to 000 by TPS End of Day process). For the EOT files accompanying manually repaired data files, the file sequence number will be 001 onwards for that day.

The detailed layouts of the BLE data file and EOT files are present in POL FS AIS [R3].

TPS will create the POL FS Summary files and EOT file in a host directory pointed to by the environment variable POLFS_OUTPUT. The physical path for this directory will be /bvnw01/tps/trans/polfs_output. Initially, this directory will be sized to be of 6 GB. (As per the sizing in the POL FS AIS [R3], each file can be up to 20MB. There will be up to 65 data files on a normal business day. The file retention period will be initially set to 5 days.)

[15] This will be same as that used by SSC for Release S60.

FUJ00090946

TPS program that creates the EOT (End of Transmission) file will also create:

- a link for the file in the audit directory pointed to by environment variable POLFS_AUDIT. The physical path for this directory will be /bvnw01/tps/trans/tps_polfs_audit. This directory will be sized to be of 6 GB. The Audit Server will copy and housekeep the files from this directory
- a copy of the file in the POL FS directory NFS mounted on the host (hard links may not span file systems). This directory will be pointed to by environment variable POLFS_SHARE. The physical path for the host directory will be /bvnw01/tps/trans/polfs_share. Size and name of the physical directory on POL FS system is outside the scope of this document.

POL FS will process and load the files present in the POLFS_SHARE directory and then delete (copies kept for diagnostic purpose) all the successfully loaded Summary files and the associated EOT file.

During the S80 migration, TPS will send "Pre-migration Summaries" and "Opening Figures" (see section 6.8) in addition to the normal POL FS Summaries. This data will be identified by separate sub-file headers and sent in a separate set of data files. Therefore, during the migration period, TPS will send more data files than that on a normal business day. It is recommended that file names for "Opening Figures" and "Pre-migration Summaries" data use a different series, for example, 201-264 for "Opening Figures" and 301-364 for "Pre-migration Summaries".

6.3.6 Reporting Incomplete POL FS Summaries

This report will list all Outlets and Trading Dates where the POL FS Summaries are found to be incomplete, i.e., the sub-file total does not balance to zero. The idea is that this report together with the Harvester Exceptions report for the day will account for all transactions harvested into TPS which have not been sent to POL FS.

The data will be extracted from POL FS Incomplete Summary table for the records with current processing date and PROCESSED flag NOT set to 'Y'. Maestro will copy this report from the host to SSC and MSU servers using the existing report copying mechanism. The report file name will be derived using the current report file naming convention.

The report is present in the Appendix D of this document.

6.3.6 Releasing the Incomplete POL FS Summaries

While generating POL FS Summary files from normal or Incomplete POL FS Summaries tables, if a pending harvester exception is found for a Branch and Trading Date, the corresponding sub-file will be held back for correction. This will be true even if a (repaired) harvester exception was not required to be included in the summary.

Any harvester exceptions will appear on the daily Harvester Exceptions Report (TPSC254) for the day. If the harvester exceptions for a Branch and Trading Day were required in the summaries, the summaries can only be released when the

FUJ00090946

harvester exceptions (required to balance the sub-file total to zero) have been repaired and the resultant summaries have been included in the POL FS Incomplete Summary table. As a part of repairing the harvester exception, the RECORD_REPAIRED flag will (automatically) be set to 'Y' in the TMS_HARVESTER_EXCEPTIONS_ARC table.

However, if the harvester exceptions for a Branch and Trading Day were not required in the summaries (for example, the transaction belonged to an excluded Mode), the summaries can be released by manually setting the RECORD_REPAIRED flag to 'Y' for these exceptions in the TMS_HARVESTER_EXCEPTIONS_ARC table.

Holding back the summaries in the above case could be avoided if it was established at the time of generating the summaries that the harvester exceptions for a Branch and Trading Day were not required to be included. But, it would be very complex to find this out, as the column values are stored in generic columns (COLUMN_NAME_n, COLUMN_VALUE_n) in the Harvester Exception tables. Moreover, it is expected that the number of harvester exceptions will be very low after relaxing the check constraints (on TPS Transaction tables) at S80, and so will be the resulting held-back-summaries.

6.4 Handling harvester exceptions due to check constraint failure(s)

At present, if an Oracle insert fails due to a check constraint violation on the TPS Transactions tables then the exception record is inserted into the TPS Harvester Exceptions table. At TPS End of Day, the process TPSC209 copies all harvester exceptions into Harvester Exceptions Archived table. The harvester exceptions are later repaired using TIP Repair Tool and stored in the 65th partition of respective Transaction tables. Refer to TPS HLD [R7] for the details.

Any repaired transaction will be summarised in the same way as the normal transactions are summarised. That is, the Initial Summaries will be stored in the 65th partition of Initial Summary table. The transactions, which are required unsummarised, may have their Account Reference Id and/or Client Reference included in the initial summary. The data from the 65th partition of Initial Summary table will be summarised and written to POL FS Incomplete Summary table.

The data in the 65th partition of Initial Summary table will also be used for HR SAP Summarisation (refer to [R14] for details).

6.5 Housekeeping the Transaction and Summary Data Tables at TPS End of Day

As mention in section 6.1.3, there will be "A" and "B" sets of Initial and POL FS Summary tables similar to those for the Transaction tables. One set will store the

FUJ00090946

current day's data and the other set will store previous day's data. Each set will have 64 partitions. For the Initial Summaries, there will be a 65th partition to store the initial summaries from repaired transactions.

At TPS End of Day, the previous and current sets of Initial and POL FS Summary tables will be swapped and then the current set truncated. The program will also truncate the 65th partition of the Initial Summary table.

Further, all records in the POL FS Incomplete Summaries table, which have been successfully written to a data file and have the "PROCESSED" flag set to 'Y', will be deleted from the table.

6.6 POL FS Summary File Rejection Handling and SLA Reporting

As present, any POL FS Summary file rejection will be dealt with manually by SSC. As implemented in S60, an alert will be raised via SMC when an ".ERR" (for a rejected POL FS file) appears in the POLFS_OUTPUT directory.

The file delivery information sent to Data Warehouse (Fujitsu Services) will be updated to include POL FS Summary data files delivery as per the updated TPS to DWh AIS [R12]. For POL FS Summaries, TPS will report on the number of originating transaction records that were aggregated to generate the POL FS Sub-files. The logical layout of the record sent to Data Warehouse is as follows:

Field Name	Value	Description
File Id	Name of the file	
Source	TPS	The source of the SLA
Dest	PFS	The destination (i.e. POL FS)
C_Date	The Creation date (EOD Date) of the sub-file	
D_Date	The delivery date/time of successful delivery of the sub-file	
Records	The number of originating transaction records that were aggregated to generate the sub-file	
FAD_Code	The FAD Code of the sub-file	

The data will be extracted from the TPS database using the following logic:

- Outer join TPS File Register (TFR) and POL FS Sub File Register (SFR) on the File_Name column. File Register will be the driving table in the join.
- Select TFR.File_Name, 'TPS' (Source), 'PFS' (Destination), SFR.Trading_Date (C_Date), TFR.Receipt_Date (D_Date), SFR.Total_Transaction_Count (Records) and SFR.Group_Id (FAD_Code) from the TPS File Register (TFR) and POL FS Sub File Register (SFR).
- Select only those records from the TPS File Register where "Destination" is set to 'P' and "File Name" is like '%BLE'.

FUJ00090946

- Select the "FAD Code" as NULL and (number of) "Records" as 0 where the file is empty, i.e., an entry is present in the TPS File Register but not in the POL FS Sub-file Register.

6.7 TPS Host Maestro Schedule Changes

TPS Maestro schedule present in the TPS Host Support Guide [R6] will be updated to run the new host programs that produce

- Initial and POL FS Summary
- POL FS Summary and EOT files

The POL FS feed for IMPACT Release 1 will become inactive at Migration Point 25 (see section 6.8.3). The actual date of migration will be a parameter driven. The feed will be removed from the Maestro schedule at Migration Point 40.

Also, the new POL FS feed for IMPACT Release 3 will become active at Migration Point 25. The new POL FS Schedule will run before the TIP/POL MIS Schedule.

A Maestro alarm will be raised if the new host program that produces the EOT file and copies the Summary/EOT files to POLFS_SHARE directory has not finished by 03:00 hours on a day.

Also, Maestro will update the existing script to copy Incomplete POLFS Summaries Report to SSC and MSU server.

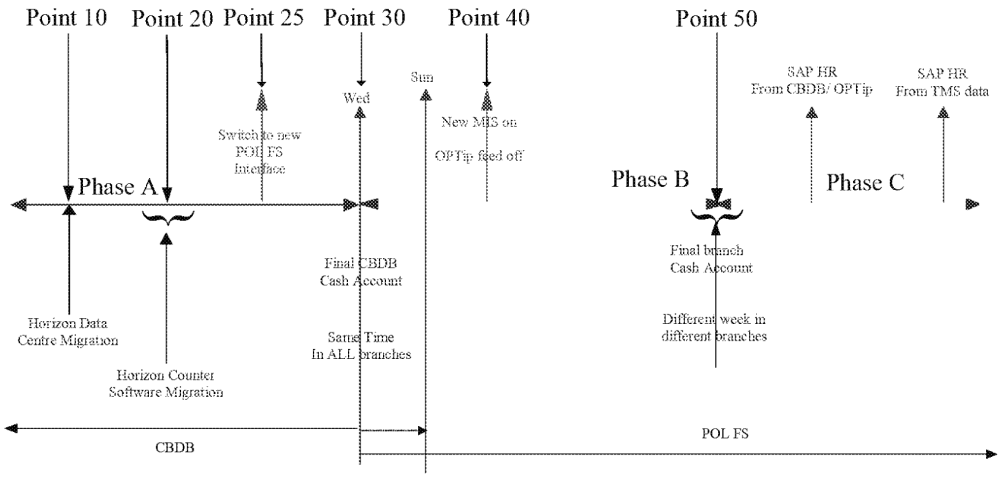
For details of the updated TPS Maestro schedule, refer to TPS Host Support Guide [R6].

6.8 Migration

The details of the migration process are present in IMPACT Release 3 Migration HLD [R15], but the relevant sections have been repeated here for convenience. Hence, in case of any discrepancies, the IMPACT Release 3 Migration HLD will take precedence.

The functionality relating to IMPACT Release 3 (Horizon Release S80) will be rolled out over a period of weeks to the Branches. It is anticipated that the Release 3 interface to POL FS will only go live once all Branches have the S80 software. Up to this point the Release 1 interface (including TPS to POL FS feed introduced at S60) will continue.

FUJ00090946



There will be three separate phases of the overall migration process:

Phase A: This is the period when the branches migrate to include support for the IMPACT Release 3 functionality. At the end of this phase, the final Cash Account is sent to TIP (to be processed by CBDB).

Phase B: This is the period when POL FS will be providing the central support for the Financial systems, however the branches will still be operating most of the current processes.

Phase C: This is when the branches switch to using Branch Trading statements rather than the current Cash Accounts.

The key points in the migration are:

- Data Centre Migration (Point 10)
- Counter Software Upgrade (Point 20)
- Switch to new POL FS Interface (Point 25)
- Running the Final Counter Cash Account for CBDB (Point 30)
- Switch of TMS feed of Transactions from OPTIP to MIS (Point 40)
- Upgrade of Counter processes to operate Branch Trading Statement (Point 50)

6.8.1 Data Centre Migration

This will be the normal upgrade process that takes place over a single weekend. It will ensure that all the Data Centre Systems (including the TPS host) are able to support the new functionality, while also retaining support for the existing functionality, prior to it being switched off.

All new or updated database and schema for TPS will be delivered at this point. Also, the new and updated application modules (along with S80 TPS Maestro schedule) will be delivered at this point.

FUJ00090946

Where required, the running of a process will be controlled via a TPS System Parameter (if the parameter is set, exit with a success without doing any processing) or Maestro until the required transactional or reference data is available.

Once this migration has taken place, it will be possible to receive the additional Reference Data required to support the new functionality from NRDS and to distribute it as required.

6.8.2 Counter Software Upgrade

This section is out of scope of TPS HLD but included here for completeness.

This will follow the normal pattern for a Software rollout and include some initial trial Branches to ensure that the process runs smoothly, prior to rolling the software out to the full estate. Some of the new functionality will become active as soon as the Branch is upgraded, while other functions will be controlled by a Soft Launch mechanism and so will be activated at a later time.

6.8.3 Switching to new POL FS Interface at S80

At migration point 25, the POL FS feed will switch to new POL FS Interface and TPS host will start summarising the transactions to produce POL FS feed. But, POL FS needs only Cash and Near Cash transactions (using S60 mappings) until final CBDB Cash Account (say 'N') is sent. Therefore, TPS will:

- summarise only Cash and Near Cash[16] transactions for CAP <= 'N' using a separate summarisation process (TPSC292) since POL FS mappings for these transactions will be those used at S60.
- summarise all[17] transactions for CAP > 'N' or CAP = Null (as CAP will not be available in future) using the new summarisation process (TPSC291) which uses the POL FS mappings at S80.

There is a requirement to keep the POL FS Summaries for CAP<='N' and CAP>'N' (or null) separate so that POL FS posts the associated transactions into different Accounting Periods. This will be achieved by including the summaries for CAP<='N' in a separate set of sub-files with a new record identifier for the sub-file header (BLCR3). For simplicity, the sub-files containing BLCR3 data will be written to separate data files, but these data files will still be included in the EOT file for the transmission.

Any summaries for CAP<='N' will require an appropriate "Balancing Transaction" to account '999999' such that the sub-files sent to POL FS balances to zero. Once a branch is fully operating at CAP>'N', then the summaries should be automatically balanced without any Balancing Transaction.

---

[16] Cash and Near Cash products will be defined in a local lookup table to avoid hard coding.

[17] If a Counter sends transactions for CAP 'N+1' before the Cash Account 'N' for the Branch is received, the new product summaries will be sent to POL FS before the Opening Figures. As agreed, this is acceptable to POL FS.

FUJ00090946

As the move from CAP 'N' to CAP 'N+1' may occur during a Trading Day as opposed to at the end of Trading Day, it is possible that data for a single Trading Day contains a mixture of transactions for CAP 'N' and CAP 'N+1'. For example, consider the following scenario:

- SU (Stock Unit) AA rolls into CAP 'N+1' at 16:00 hours on Tuesday (since the owner of the SU doesn't normally work on Wednesday).
- SU AA is then used at 17:00 on Tuesday (due to a late rush at closing time)
- SU BB is rolled over at 16:00 on Wednesday (as normal)
- SU CC is rolled over at 09:30 on Thursday morning (after entering out of hours transactions – e.g. for lottery)
- Cash Account is produced at 10:00 on Thursday morning (as normal)

In this case there will be transactions for both CAP 'N' and CAP 'N+1' on Tuesday, Wednesday and Thursday and the Cash Account will not be available for processing until Thursday evening.

During the migration period, when a mixture of transactions for CAP <= 'N' or CAP>'N' are received, the value of Sub-file Id Flag will be either set to "BLCR1" or "BLCR3". After the migration period, i.e., for CAP > 'N' or CAP = Null (as CAP will not be available in future), the value of Sub-file Id Flag will always be set to "BLCR1".

On the Trading Days which have a mixture of transactions for CAP 'N' and CAP 'N+1', the POL FS Summary processing will be as follows:

- TPSC292 will summarise Initial Summaries where CAP<='N' using the mappings available in TPS_POL_FS_MAPPINGS_AT_S60. These locally defined mappings will be based on POL FS mappings in *Mapping of Horizon products to POL FS Chart of Accounts Codes* (EA/CDE/001) and will be available for Cash and Near Cash products only. TPSC292 will store these summaries in the POL_FS_SUMMARIES table with Sub-file Id Flag set to "BLCR3". When written to POL FS data file, the sub-files will contain a Balancing Transaction to account '999999' such that the BLCR3 sub-file totals balance to zero.
- TPSC291 will summarise Initial Summaries where CAP>'N' using the S80 mappings available in TPS_POL_FS_MAPPINGS. At S80, the POL FS Summaries mapping will be based on the Reference Data supplied by POL's NRDS system and available for all products. TPSC291 will store these summaries in the POL_FS_SUMMARIES table with Sub-file Id Flag set to "BLCR1". When written to POL FS data file, these summaries will have sub-files with identifier BLCR1. The sub-files will **not** contain a Balancing Transaction.

The details of TPSC291 and TPSC292 processes are in the sections 6.2.2 and 6.2.3 respectively.

The current S60 POL FS feed will be stopped at migration point 25. The existing processes TPSC274 and TPSC275 will be updated to achieve this (see section 7.1 for details).

FUJ00090946

6.8.4 Running the Final Counter Cash Account for CBDB

It is a requirement of the POL FS designers that the switch over of the accounts from CBDB to POL FS occurs at a single point in time, which coincides with a POL Month End. A specific Cash Account Week will be identified such that once that Cash Account has been produced, all subsequent transactions will be summarised (in TPS host) and passed to POL FS.

In addition, a special migration flow of data (produced in the TPS host) will be required to pass the Closing Figures from that Cash Account through to POL FS as Opening Figures for the corresponding accounts. All transactions from the point at which the Final cash account was taken must be identified and their effect passed to POL FS even if they took place in earlier Trading Days so that all Transactions are accounted for in either CBDB or POL FS.

POL FS requires opening figures for all Stock products, plus the initial position for all suspense and discrepancy items. No opening figures are required for [CP3884 requires opening figures for Cash and Cash in Pouches to be sent] cheques since POL FS will already have this data from Impact R1.

There will be a separate process TPSC293 (which will be thrown away after the migration) to generate the POL FS Opening Figures from the Counter Cash Account (CAC) and Stock Holding (STX) records. The details of TPSC293 process are in the sections 6.2.4.

6.8.5 Switch of TMS feed of Transactions from OPTIP to MIS

This section is covered by TPS HR SAP Summarisation & Transaction Correction HLD [R14] but included here for completeness.

Once all Cash Account data from the final cash account has been successfully passed to OPTIP, the Transactional flow (from TPS host) to OPTIP will be discontinued and replaced by the new flow to MIS.

However, during the Phase B there will still be cash account information coming from some branches (following non-polling) which will need to be sent to CBDB. In order to support this, the existing Cash Account to OPTIP will need to be maintained during this period. Once all Final Cash Accounts for a Branch have been sent through, it will then be possible to switch off the feed to OPTIP and to replace it with an enhanced data feed to MIS. The final (CBDB) Cash Account will be stored as a TPS System Parameter.

At the time of switching off the Transaction Feed to TIP, a number of database check constraints (e.g. CAP and Balance Period being not null; full set to be defined) will be removed from the TPS Transaction tables, as the new MIS system does not validate these items.

This includes the transactions where the physical closing of the Final Cash Account occurred after the Wednesday night. This will ensure that no transaction is accounted for in both CBDB and POL FS – double counting.

FUJ00090946

6.8.6 Upgrade of Counter processes to operate Branch Trading Statement

This section is out of scope of TPS HLD but included here for completeness.

At this point in time, the branches will switch to using the new processes to operate Branch Trading Statement.

More details of overall migration process are in IMPACT Release 3 Migration HLD [R15].

FUJ00090946

7. TPS Host Application, Database and Schema Changes

7.1 TPS Application Changes

New binaries will be delivered for the new processes populating Initial and POL FS Summary tables and producing POL FS Summary and EOT files.

All programs will start with logging the control information in TPS_PROCESS_CONTROL and TPS_PROCESS_AUDIT tables. The existing TPS common functions will be used to perform these activities.

7.1.1 TPSC207 - Copy Reference Data

The existing TPS Reference Data Copy program will be modified to copy the additional reference data required for POL FS Summarisation (see section 6.1.1).

This program will also copy reference data required for HR SAP Feed (see [R14]).

As the data might not be available in RDDS table immediately at migration point 10, the program should continue if it finds no records to be copied for the newly added reference data tables, e.g., TPS_POL_FS_MAPPINGS table.

Migration Point 25 onwards, POL FS reference data for the day must be copied into TPS to get and use the latest mappings. The program, therefore, should fail if it can not copy POL FS reference data for the day.

As a part of CP3844, a new column 'Site Code' will be added to the TPS_OUTLETS table. TPSC207 must be updated to copy this new column into TPS database. As the details of other new columns added to TPS_OUTLETS table at S80 are in TPS HR SAP and Transaction Correction HLD [R14], this document will be updated to add 'Site Code' to the TPS_OUTLETS table. The up-to-date definition of the TPS_OUTLETS table is also present in RDDS to TPS AIS [R13].

7.1.2 TPSC290 - Generate Initial Summary

A new TPS program will generate the initial summaries from the transactions and populate Initial Summary tables. Multiple instances of the program will be run in parallel to generate the Initial Summaries.

Details of this process are present in the section 6.2.1.

FUJ00090946

7.1.3 TPSC291 - Generate POL FS Summary (Final CBDB CAP onwards)

A new TPS program will generate the POL FS summaries from the Initial Summaries where CAP is greater than *<Final CBDB CAP>* and populate POL FS Summary tables. The summary records will have Sub-file Id Flag set to "BLCR1". Multiple instances of the program will be run in parallel to generate the POL FS Summaries.

Details of this process are present in the section 6.2.2.

7.1.4 TPSC292 - Generate POL FS Summary (Before Final CBDB CAP)

A new TPS program will generate the POL FS summaries from the Initial Summaries where CAP is less than or equal to *<Final CBDB CAP>* and populate POL FS Summary tables. The summary records will have Sub-file Id Flag set to "BLCR3". Multiple instances of the program will be run in parallel to generate the POL FS Summaries.

Once all branches have produced the Final CBDB CAP, i.e., no transactions are expected for a CAP <= *<Final CBDB CAP>*, this program can be switched off (this can be done at migration point 40).

Details of this process are present in the section 6.2.3.

7.1.5 TPSC293 - Produce Opening Figures for POL FS Summaries (at Final CBDB CAP)

A new program will produce the Opening Figures for the POL FS. This program will be required only at the initial stage of Impact Release 3. The corresponding summary records will have Sub-file Id Flag set to "BLCR2". Multiple instances of the program will be run in parallel to generate the POL FS Summaries.

Once all branches have produced the Final CBDB CAP, i.e., no CAC or STX records are expected for a CAP <= *<Final CBDB CAP>*, this program can be switched off (this can be done at migration point 40).

Details of this process are present in the section 6.2.4.

7.1.6 TPSC294 - Produce POL FS Summary Files

A new program[18] will write the POL FS summaries to a set of data files as per the new format in POL FS AIS [R3]. Multiple instances of the program will be run in parallel to produce the data files.

---

[18] It has been decided to have a new program instead of updating the existing S60 program because:
- if we have a migration rehearsal we will need to run both the programs in parallel.
- The processing logic, summary table and record format are significantly different.

FUJ00090946

The program will create an entry in TPS File Register for each data file produced. The new column DESTINATION added to TPS File Register will be set to 'P' for the data file. Also, the program will create an entry in the POL FS Sub-file Register for each complete sub-file produced. The entries for any incomplete sub-files will be created when these sub-files become complete and are actually sent to POL FS.

As defined in POL FS AIS [R3], the Label Identifier in the Branch Ledger Entry Contents record will be set as per the value of Sub-file Id Flag in the TPS_POL_FS_SUMMRIES table.

Details of this process are present in the section 6.3.

7.1.7 TPSC295 - Produce EOT File

A new program will create an EOT (End of Transmission) file once all the POL FS Summary files have been produced, as per the format in POL FS AIS [R3]. The EOT files will contain the file names of the Summary files for that day.

The program will create an entry in TPS File Register for each EOT file produced. The new column DESTINATION added to TPS File Register will be set to 'P' for the data file.

This program will also create links/copies of the Summary/EOT files in the POLFS_AUDIT and POLFS_SHARE directories and record the date and time of copy in the TPS File Register table.

The program will raise an alert if a previous EOT file is present in POLFS_SHARE directory, as at S60.

Details of this process are present in the section 6.3.

7.1.8 TPSC206 - Update Outward File Delivery Information for Data Warehouse

This HLD covers the changes related to POL FS Feed only. The other changes related to HR SAP Feed, etc. will be covered by [R14].

The existing Produce Outward Delivery File Summary program will be modified to send delivery information of POL FS Summary data files.

Details of this change are present in the section 6.6.

7.1.9 TPSC209 - Housekeeping/Truncating Summaries tables at TPS EOD

The existing TPS End of Day program will be modified to housekeep the new Initial and POL FS Summary tables.

[DN: Deleted the above two paragraphs, as these were already present in section 6.5.]

Details of the changes to this process are present in the section 6.5.

FUJ00090946

7.1.10 TPSC257 - POL FS Incomplete Summaries Report

A new program will create a report containing all Outlets and Trading Dates where the POL FS Summaries have been found to be incomplete, i.e., the sub-file totals did not balance to zero.

Details of this process are present in the section 6.3.6.

7.1.11 TPSC274 and TPSC275 - POL FS Summaries at S60

These programs will be updated to exit without any processing if the TPS System Parameter "S80 MIGRATION POINT" is greater than or equal to 25.

7.2 TPS Database and Schema Changes

For the additional transaction volumes and new functionality introduced at S80, additional disk space and Oracle objects will be required in the TPS database. A set of database and schema build scripts will be delivered to add the disk space and objects required for POL FS Summaries.

Details of any additional disk space required will be present in the TPS Database Sizing spreadsheet.

The COFA Transaction and Summary tables created in TPS database at S60 may be dropped once the TPS has switched from POL FS Feed Release 1 to 3. The associated views, public and private synonyms may also be dropped at the same time. But, it would be risky to drop a large number of objects from live database in the middle of a release. It is therefore recommended that these (empty) objects be dropped post S80.

7.2.1 TPS Performance & Volume at S80

At present, the Agents do not harvest all the transactions into TPS host. The settlement transactions with products in the range 11000 to 20000 are not required to be passed to TIP and therefore not harvested into TPS host. However, the volumes of such transactions are negligible compared with the rest. Therefore, these transactions will not have an effect on the Transaction table sizing.

As from S80 onwards, all transactions will be required in TPS host to balance the sub-file totals to zero, the database sizing requirements need to be revisited. Further, there will be storage required for the additional attributes[19] ("TC Reference", "Pouch Id" and "Client Reference") in the EPOSS Transactions, and Initial, POL FS and HR SAP Summaries. The existing TPS data tablespaces TPS_PARTITON_0 to TPS_PARTITION_15 will be expanded to store the additional transactions and summaries.

As defined in [R16], the new transaction volumes and design limits for TPS will be:

[19] The estimated volume of transactions including these attributes will be negligible. So, no effects on the Transaction table sizing.

FUJ00090946

Table Type	Message Type	Volume[20]	Contracted Volume	Design Limit (20% higher)
Transaction Tables	Settlement[21]	Peak 2 Days	17,188,486	20,626,183
		Peak Day	9,565,842	11,479,010
		Subtotal	**26,754,328**	**32,105,193**
	EPOSS	Peak 2 Days	15,620,323	18,744,387
		Peak Day	8,602,518	10,323,021
		Subtotal	**24,222,841**	**29,067,409**
	APS	Peak 2 Days	5,626,222	6,751,466
		Peak Day	3,031,573	3,637,887
		Subtotal	**8,657,795**	**10,389,354**
	OBCS[22]	Peak 2 Days	8,144,758/2 = 4,072,379	9,773,710/2 = 4,886,855
		Peak Day	4,718,625 = 2,359,312	5,662,350 = 2,831,175
		Subtotal	**6,431,691**	**7,218,030**
	NWB	Peak 2 Days	5,656,759	6,788,111
		Peak Day	3,264,181	3,917,017
		Subtotal	**8,920,940**	**10,705,128**
	DCS	Peak 2 Days	565,949	679,139
		Peak Day	288,425	346,110
		Subtotal	**854,374**	**1,025,248**
	Bureau	Peak 2 Days	135,567	162,680
		Peak Day	76,775	92,130
		Subtotal	**212,342**	**254,810**
Initial Summary	N/A	Peak Day x 2	5,440,000[23]	6,524,000
POL FS Summary	N/A	Peak Day x 2	5,440,000	6,524,000

[20] The database has been sized so that it can handle Peak 2 Days' transactions harvested on a single day. Also, it should be able to keep one Peak Day's transactions in the saved set of tables.

[21] TPS Harvester currently puts most of these settlement transactions (for example, tendering cash for serve customer transaction) into EPOSS transaction tables. The additional "dummy" settlement transactions harvested at S80 are estimated to be less than 100,000 a day So, no effects on the Transaction table sizing.

[22] By S80, the OBCS transaction volume should have got down to the NBS fully rollout volumes (i.e. peak day of 235K; see [R16]). But, it will be safer to use the 50% of the rolled out numbers (i.e. design limit of 2.8M peak day). So, the unused space can be used for any additional Transactions harvested into TPS at S80.

[23] Based on the estimate of 171 detail lines per branch in the POL FS AIS [R3], the POL FS Summary volumes for 16000 branches will be approximately 2.72 million records per day. The volumes of Initial Summary will therefore be more since the data is not summarised by Article at this level. But, this figure of 171 is based on number of different products transacted at the Counter and considered to be on the higher side for POL FS Summaries. It has therefore been decided to size both POL FS and Initial Summaries to be approximately 2.72 million records per day. Also, the table for POL FS and Initial Summaries will share the same tablespace so that space is utilised efficiently if one has less volume and one has more.

FUJ00090946

Other Intermediate and Static Data Storage	N/A	N/A	As per the current set up.	As per the current set up

The Transaction and Summary tables will be sized to store up to 2 Peak Days of transaction volume. Further, there will be a saved set of Transaction and Summary tables sized to store up to a Peak Day of transaction volume. So, the total storage is for up to 3 Peak Days of transaction volume.

7.2.2 TPS Schema Changes

The details of TPS schema changes related to POL FS Summaries are present in Appendix A-C of this document.

FUJ00090946

Appendix A - Table and Index Definitions

Table Definitions

Modification to Existing Tables

i. TMS_RX_EPOSS_TRANSACTIONS_xxxx

These tables store the EPOSS Transactions harvested into TPS. The table name contains 'xxxx' to denote the partition number and table set. For example, 'xxxx' is in "1A, 2A, .., 64A, 1B, 2B, .., and 64B". The following new column(s) will be added to these tables:

```
Column Name                    Null?    Type
-----------------------------  -------- ----
POUCH_ID                                VARCHAR2(32)
TC_REFERENCE                            VARCHAR2(32)
CLIENT_REFERENCE                        VARCHAR2(16)
```

The above columns will also be added to the TMS_RX_EPOSS_TRANSACTIONS_65/65RC tables.

ii. TMS_RX_STOCK_HOLDINGS_xxxx

These tables store the Stock Holding records harvested into TPS. The table name contains 'xxxx' to denote the partition number and table set. For example, 'xxxx' is in "1A, 2A, .., 64A, 1B, 2B, .., and 64B". The following new column(s) will be added to these tables:

```
Column Name                    Null?    Type
-----------------------------  -------- ----
PURCHASED_QUANTITY                      NUMBER(14)
```

iii. TMS_HARVESTER_EXCEPTIONS_xx

These tables store the TPS harvester exceptions. The table name contains 'xx' to denote the partition number ranging from 1 to 64. The following new column(s) will be added to these tables:

```
Column Name                    Null?    Type
-----------------------------  -------- ----
GROUP_ID                                NUMBER(6)
```

FUJ00090946

iv. TMS_HARVESTER_EXCEPTIONS_ARC

All harvester exceptions are copied into this table at TPS End of Day. The following new column(s) will be added to this tables:

```
Column Name                    Null?    Type
------------------------------ -------- ----
GROUP_ID                                NUMBER(6)
```

New Tables

i. TPS_PROD_MODE_SUMMARIES_xxxx

These tables will store the initial transaction summaries by Product and Transaction Mode. The table name contains 'xxxx' to denote the partition number and table set. For example, 'xxxx' is in "1A, 2A, .., 64A, 1B, 2B, .., and 64B". The two table sets "A" and "B" store the current day's and previous day's data.

Two new sets of these tables will be created to store Initial Summaries. Each table set will have 64 partitions. The partition numbering will be same as that for TPS Transaction tables. These tables will be accessed via views similar to Transaction tables.

There will be a 65th partition of the table to store the Initial Summaries generated from repaired transactions present in the 65th partition of Transaction tables. This table will be accessed via a public synonym. As mentioned in section 6.1.3, there is no need to have an archive ("65RC") table associated with the 65th partition of the Initial Summary table.

The "Amount" column is defined as Number(9, 2) in the base transaction tables. Also, the final total of transaction amounts, i.e., the "Account Value" is defined as Number(9, 2) in the POL FS AIS [R3] Therefore, the intermediate total "Total Transaction Value" column has been defined as Number(9, 2).

The Quantity[24] column is currently (pre S80) defined as Number(5) but will change to Number(14) in the transaction tables. Also, the final total of the transaction quantities, i.e., the "Account Quantity" is defined as Number(15) in POL FS AIS [R3]. Therefore, the intermediate total "Total Transaction Quantity" column has been defined as Number(15).

The physical layout of the table is as follows:

```
Column Name                    Null?    Type
------------------------------ -------- ----
TRADING_DATE                   NOT NULL DATE
GROUP_ID                       NOT NULL NUMBER(6)
PROD_ID                        NOT NULL NUMBER(10)
TRANSACTION_MODE_ID            NOT NULL NUMBER(2)
TOTAL_TRANSACTION_QUANTITY     NOT NULL NUMBER(15)
TOTAL_TRANSACTION_AMOUNT       NOT NULL NUMBER(9,2)
```

[24] For Bureau Transactions the Quantity value needs to come from the "PURCHASED_QUANTITY" column, not the "QUANTITY" column.

FUJ00090946


```
TOTAL_TRANSACTION_COUNT          NOT NULL NUMBER(6)
POUCH_ID                                  VARCHAR2(32)
TC_REFERENCE                              VARCHAR2(32)
CLIENT_REFERENCE                          VARCHAR2(16)
CASH_ACCOUNT_PERIOD                       NUMBER(2)
INSERT_DATE                      NOT NULL DATE DEFAULT SYSDATE
```

These not null constraints have been implemented as per the mandatory data fields present in TPS Transaction tables. When mapped to POL FS Summaries, some of these fields will be required only if the data is being sent unsummarised. Therefore, these fields have been defined as optional in the POL FS AIS [R3].

The CASH_ACCOUNT_PERIOD is required for migration period only. During migration, the summarisation process will take the final CBDB CAP into account while generating the summaries.

The INSERT_DATE is for diagnostic purpose.

ii. TPS_POL_FS_SUMMARIES_xxxx

These tables will store the POL FS summaries. The table name contains 'xxxx' to denote the partition number and table set. For example, 'xxxx' is in "1A, 2A, .., 64A, 1B, 2B, .., and 64B". The two table sets "A" and "B" store the current day's and previous day's data.

Two new sets of these tables will be created to store POL FS Summaries. Each table set will have 64 partitions. The partition numbering will be the same as that for TPS Transaction tables. These tables will be accessed via views similar to Transaction tables.

There will be a separate partition of the table (TPS_POL_FS_SUMMARIES_INCOMP) to store the Incomplete Summaries, i.e., the POL FS summaries generated from the 65th partition of Initial Summary table or the POL FS Summaries where the net total for the Branch and Trading Date does not balance to zero. This table will have two additional columns, "PROCESSED" flag to indicate that the summaries have now been corrected and sent to POL FS, and TPS_SYTEM_DATE to record the TPS Processing Date. This flag will be used by TPS Housekeeping process to delete the processed records from the table. The Processing Date will be used in the POL FS Incomplete Summaries Report. This is the logical date TPS would have sent the summaries to POL FS had these summaries balanced. The table will be accessed via a public synonym.

The TRANSACTION_MODE_ACRONYM is the acronym (e.g. SC, DDP, DDN, RIAD, ROAD, etc.) for TRANSACTION_MODE_ID (e.g. 1, 17, 19, 23, 25, etc.). The POL FS Summary records written to the data files will contain TRANSACTION_MODE_ACRONYM.

FUJ00090946

The physical layout of the table is as follows:

```
Column Name                    Null?    Type
------------------------------ -------- ----
TRADING_DATE                   NOT NULL DATE
GROUP_ID                       NOT NULL NUMBER(6)
ARTICLE_ID                     NOT NULL VARCHAR2(10)
ARTICLE_TYPE                            NOT NULL VARCHAR2(2)
TRANSACTION_MODE_ID                     NUMBER(2)
TRANSACTION_MODE_ACRONYM                VARCHAR2(10)
SETTLEMENT                              VARCHAR2(1)
LEDGER                                  VARCHAR2(1)
ACCOUNT                                 NUMBER(10)
MOVEMENT_TYPE                           NUMBER(3)
ACCOUNT_VALUE                           NUMBER(9,2)
ACCOUNT_QUANTITY                        NUMBER(15)
ACCOUNT_REFERENCE_ID                    VARCHAR2(32)
CLIENT_REFERENCE                        VARCHAR2(16)
TOTAL_TRANSACTION_COUNT        NOT NULL NUMBER(8)
SUB_FILE_ID_FLAG               NOT NULL VARCHAR2(5)
INSERT_DATE                    NOT NULL DATE DEFAULT SYSDATE
```

The physical layout of the table POL FS Incomplete Summaries table is as follows:

TPS_POL_FS_SUMMARIES_INCOMP

```
TRADING_DATE                   NOT NULL DATE
GROUP_ID                       NOT NULL NUMBER(6)
ARTICLE_ID                     NOT NULL VARCHAR2(10)
ARTICLE_TYPE                            NOT NULL VARCHAR2(2)
TRANSACTION_MODE_ID                     NUMBER(2)
TRANSACTION_MODE_ACRONYM                VARCHAR2(10)
SETTLEMENT                              VARCHAR2(1)
LEDGER                                  VARCHAR2(1)
ACCOUNT                                 NUMBER(10)
MOVEMENT_TYPE                           NUMBER(3)
ACCOUNT_VALUE                           NUMBER(9,2)
ACCOUNT_QUANTITY                        NUMBER(15)
ACCOUNT_REFERENCE_ID                    VARCHAR2(32)
CLIENT_REFERENCE                        VARCHAR2(16)
TOTAL_TRANSACTION_COUNT        NOT NULL NUMBER(8)
SUB_FILE_ID_FLAG               NOT NULL VARCHAR2(5)
INSERT_DATE                    NOT NULL DATE DEFAULT SYSDATE
INSERT_SYSTEM_DATE             NOT NULL DATE
PROCESSED                               VARCHAR2(1)
```

These check constraints have been implemented as per the mandatory data fields present in POL FS AIS [R3].

The INSERT_DATE is for diagnostic purpose.

iii. TPS_POL_FS_ARTICLES

This table will be used to copy POL FS Article details from the RDDS database.

```
Column Name                    Null?    Type
------------------------------ -------- ----
ARTICLE_ID                     NOT NULL VARCHAR2(10)
DESCRIPTION                    NOT NULL VARCHAR2(30)
ARTICLE_TYPE                            NOT NULL VARCHAR2(2)
```

FUJ00090946


```
DEFAULT_PROD_POS                    NUMBER(10)
DEFAULT_PROD_NEG                    NUMBER(10)
ARTICLE_SUMMARISATION               VARCHAR2(1)
QUANTITY                            VARCHAR2(2)
```

The column ARTICLE_ID will be the primary key of the table.

iv. TPS_POL_FS_ACCOUNTS

This table will be used to copy POL FS Account details from the RDDS database. This table definition is available in [R14] but included here for completeness.

```
Column Name                    Null?    Type
------------------------------ -------- ----
ACCOUNT                        NOT NULL NUMBER(10)
DESCRIPTION                    NOT NULL VARCHAR2(50)
LEDGER                         NOT NULL VARCHAR2(1)
SETTLEMENT                              VARCHAR2(1)
```

The column ACCOUNT will be the primary key of the table.

v. TPS_POL_FS_MAPPINGS

This table will be used to copy POL FS Article and Movement Type mappings from the RDDS database.

The table will contain a default mapping for each Product to Article and Movement Type mapping. This default mapping will have a Transaction Mode value of zero. This default is to be used when no mapping for a specific Transaction Mode exists for the Product. However, such default will only exist if there is no mapping with a TRANSACTION_MODE_ID.

```
Column Name                    Null?    Type
------------------------------ -------- ----
PROD_ID                        NOT NULL NUMBER(10)
TRANSACTION_MODE_ID            NOT NULL NUMBER(10)[25]
START_DATE                     NOT NULL DATE
END_DATE                                DATE
ARTICLE_ID                     NOT NULL VARCHAR2(10)
ARTICLE_TYPE                               NOT NULL VARCHAR2(2)
QUANTITY                       NOT NULL VARCHAR2(2)
SUMMARISATION                  NOT NULL VARCHAR2(1)
REFERENCE                      NOT NULL VARCHAR2(2)
MOVEMENT_TYPE                           NUMBER(3)
ACCOUNT                                 NUMBER(10)
SETTLEMENT                              VARCHAR2(1)
LEDGER                                  VARCHAR2(1)
```

The columns PROD_ID + TRANSACTION_MODE_ID + START_DATE will be the primary key of the table.

[25] The corresponding column name in RDDS is TRANS_MODE_ID. Also, this column is currently defined as Number(2) in the Transaction tables.

FUJ00090946

vi. TPS_POL_FS_MAPPINGS_AT_S60

This table will store locally defined mappings based on POL FS mappings in *Mapping of Horizon products to POL FS Chart of Accounts Codes* (EA/CDE/001) which are available for Cash and Near Cash products only.

Having a separate table for these mappings will help if some column values are not required, hence a relaxed set of check constraints is implemented for the S60 mappings.

The mappings will be used for POL FS Summaries generated by TPSC292 for CAP <= Final CBDB CAP.

```
Column Name                    Null?    Type
------------------------------ -------- ----
PROD_ID                        NOT NULL NUMBER(10)
TRANSACTION_MODE_ID            NOT NULL NUMBER(10)
START_DATE                     NOT NULL DATE
END_DATE                                DATE
ARTICLE_ID                     NOT NULL VARCHAR2(10)
ARTICLE_TYPE                               NOT NULL VARCHAR2(2)
QUANTITY                       NOT NULL VARCHAR2(2)
SUMMARISATION                  NOT NULL VARCHAR2(1)
REFERENCE                      NOT NULL VARCHAR2(2)
MOVEMENT_TYPE                           NUMBER(3)
ACCOUNT                                 NUMBER(10)
SETTLEMENT                              VARCHAR2(1)
LEDGER                                  VARCHAR2(1)
```

The columns PROD_ID + TRANSACTION_MODE_ID + START_DATE will be the primary key of the table.

vii. TPS_TRANS_MODE_CONVERSIONS

This table will have the numeric Post Office transaction modes to alphabetic Fujitsu Services transaction modes mappings.

```
Column Name                    Null?    Type
------------------------------ -------- ----
POCL_TRANS_MODE_TYPE_CODE      NOT NULL NUMBER(10)
PATHWAY_TRANS_MODE_TYPE_CODE            VARCHAR2(10)
```

The column POCL_TRANS_MODE_TYPE_CODE will be the primary key of the table.

viii. TPS_POL_FS_SUB_FILE_REGISTER

```
FILE_NAME                      NOT NULL VARCHAR2(200)
GROUP_ID                       NOT NULL NUMBER(6)
TRADING_DATE                   NOT NULL DATE
SUB_FILE_ID_FLAG               NOT NULL VARCHAR2(5)
SUB_FILE_SEQUENCE_NUMBER       NOT NULL NUMBER(5)
TOTAL_RECORDS                  NOT NULL NUMBER(7)
TOTAL_TRANSACTION_COUNT        NOT NULL NUMBER(8)
SLA_REPORTED                            VARCHAR2(1)
PROCESS_NAME                            VARCHAR2(50)
SYSTEM_DATE                             DATE
PARTITION_NUMBER                        NUMBER(3)
INSERT_DATE                    NOT NULL DATE
```

FUJ00090946

ix. TPS_POL_FS_EXCLUDED_MODES

This table will have Transaction Mode to be excluded from POL FS feed.

```
TRANSACTION_MODE_ID             NOT NULL NUMBER(10)
```

x. TPS_OPENFIG_CAC_PROD_MAPPINGS

This table will have Cash Account Line to Product Id mappings for Discrepancy and Suspense Products used in generating the Opening Figures for POL FS Summaries.

```
LINE_NUMBER                     NOT NULL NUMBER(4)
PROD_ID                         NOT NULL NUMBER(10)
ACCOUNTING_SENSE                NOT NULL NUMBER(1)
```

Index Definitions

No indexes will be required for Initial or POL FS Summary tables.

The tables used to store Reference Data will have the following primary key indexes to enforce uniqueness, so that the joins do not find duplicate rows while deriving the POL FS summaries:

i. TPS_POL_FS_ARTICLES_PK

This primary key index will be on following columns of the TPS_POL_FS_ARTICLES table:

```
Column Name                     Null?    Type
------------------------------- -------- ----
ARTICLE_ID                      NOT NULL VARCHAR2(10)
```

ii. TPS_POL_FS_ACCOUNTS_PK

This primary key index will be on following columns of the TPS_POL_FS_ACCOUNTS table

```
Column Name                     Null?    Type
------------------------------- -------- ----
ACCOUNT                         NOT NULL NUMBER(10)
```

iii. TPS_POL_FS_MAPPINGS_PK

This primary key index will be on following columns of TPS_POL_FS_MAPPINGS table:

```
Column Name                     Null?    Type
------------------------------- -------- ----
PROD_ID                         NOT NULL NUMBER(10)
TRANSACTION_MODE_ID             NOT NULL NUMBER(10)
START_DATE                      NOT NULL DATE
```

FUJ00090946

iv. TPS_POL_FS_MAPPINGS_AT_S60_PK

This primary key index will be on following columns of TPS_POL_FS_MAPPINGS_AT_S60 table:

```
Column Name                    Null?    Type
------------------------------ -------- ----
PROD_ID                        NOT NULL NUMBER(10)
TRANSACTION_MODE_ID            NOT NULL NUMBER(10)
START_DATE                     NOT NULL DATE
```

v. TPS_TRANS_MODE_CONVERSIONS_PK

This primary key index will be on following columns of TPS_TRANS_MODE_CONVERSIONS table:

```
Column Name                    Null?    Type
------------------------------ -------- ----
POCL_TRANS_MODE_TYPE_CODE      NOT NULL NUMBER(10)
```

vi. TPS_POL_FS_SUB_FILE_REG_PK

This primary key index will be on following columns of TPS_POL_FS_SUB_FILE_REGISTER table:

```
Column Name                    Null?    Type
------------------------------ -------- ----
GROUP_ID                       NOT NULL NUMBER(6)
TRADING_DATE                   NOT NULL DATE
SUB_FILE_ID_FLAG               NOT NULL VARCHAR2(5)
```

The Sub_file_Id_Flag is included in the primary key because there could be BLCR3 (pre-migration) and BLCR2 (opening figures) type sub-files for the same Branch and Trading Date.

FUJ00090946

Appendix B - View and Synonym Definitions

View Definitions

i. Views on TPS_PROD_MODE_SUMMARIES_xxxx tables

Two new sets of views will be created to point to the partitions in "Current" and "Saved" of summary tables. Each table set will have 64 views.

The view names and definitions will be similar to those for TPS Transaction tables. The table below shows a few examples of the view names and their definitions:

View Name	View Definition
TPS_PROD_MODE_SUMMARIES_1	SELECT * FROM TPS_PROD_MODE_SUMMARIES_1A
TPS_PROD_MODE_SUMMARIES_2	SELECT * FROM TPS_PROD_MODE_SUMMARIES_2A
.. .. ..	
TPS_PROD_MODE_SUMMARIES_64	SELECT * FROM TPS_PROD_MODE_SUMMARIES_64A
SAV_ PROD_MODE_SUMMARIES _1	SELECT * FROM TPS_PROD_MODE_SUMMARIES_1B
SAV_ PROD_MODE_SUMMARIES _2	SELECT * FROM TPS_PROD_MODE_SUMMARIES_2B
.. .. ..	
SAV_ PROD_MODE_SUMMARIES _64	SELECT * FROM TPS_PROD_MODE_SUMMARIES_64B

ii. Views on TPS_POL_FS_SUMMARIES_xxxx tables

Two new sets of views will be created to point to the partitions in "Current" and "Saved" of tables. Each table set will have 64 views.

The view names and definitions will be similar to those for Transaction tables. The table below shows a few examples of the view names and their definitions:

View Name	View Definition
TPS_POL_FS_SUMMARIES_1	SELECT * FROM TPS_POL_FS_SUMMARIES_1A
TPS_POL_FS_SUMMARIES_2	SELECT * FROM TPS_POL_FS_SUMMARIES_2A
.. .. ..	
TPS_POL_FS_SUMMARIES_64	SELECT * FROM TPS_POL_FS_SUMMARIES_64A
SAV_ POL_FS_SUMMARIES_1	SELECT * FROM TPS_POL_FS_SUMMARIES_1B
SAV_ POL_FS_SUMMARIES_2	SELECT * FROM TPS_POL_FS_SUMMARIES_2B
.. .. ..	
SAV_ POL_FS_SUMMARIES_64	SELECT * FROM TPS_POL_FS_SUMMARIES_64B

FUJ00090946

iii. Overall view on TPS_PROD_MODE_SUMMARIES_xxxx tables

There will an overall view on the TPS_PROD_MODE_SUMMARIES_xx views.

The view name and definition will be similar to those for TPS Transaction tables, that is:

```
CREATE OR REPLACE VIEW TPS_PROD_MODE_SUMMARIES AS
SELECT tb.*,ROWID rid,1 Part FROM TPS_PROD_MODE_SUMMARIES_1 tb
UNION ALL SELECT tb.*,ROWID rid,2 Part FROM TPS_PROD_MODE_SUMMARIES_2 tb
UNION ALL SELECT tb.*,ROWID rid,3 Part FROM TPS_PROD_MODE_SUMMARIES_3 tb
UNION ALL SELECT tb.*,ROWID rid,4 Part FROM TPS_PROD_MODE_SUMMARIES_4 tb
.. .. ..
UNION ALL SELECT tb.*,ROWID rid,64 Part FROM TPS_PROD_MODE_SUMMARIES_64 tb;
```

Also, there will a similar overall view on the SAV_PROD_MODE_SUMMARIES_xx views.

iv. Overall view on TPS_POL_FS_SUMMARIES_xxxx tables

There will an overall view on the TPS_POL_FS_SUMMARIES_xx tables.

The view name and definition will be similar to those for TPS Transaction tables, that is:

```
CREATE OR REPLACE VIEW TPS_POL_FS_SUMMARIES AS
SELECT tb.*,ROWID rid,1 Part FROM TPS_POL_FS_SUMMARIES_1 tb
UNION ALL SELECT tb.*,ROWID rid,2 Part FROM TPS_POL_FS_SUMMARIES_2 tb
UNION ALL SELECT tb.*,ROWID rid,3 Part FROM TPS_POL_FS_SUMMARIES_3 tb
UNION ALL SELECT tb.*,ROWID rid,4 Part FROM TPS_POL_FS_SUMMARIES_4 tb
.. .. ..
UNION ALL SELECT tb.*,ROWID rid,64 Part FROM TPS_POL_FS_SUMMARIES_64 tb;
```

Also, there will a similar overall view on the SAV_ POL_FS_SUMMARIES_xx views.

v. Overall view on TPS_HARVESTER_EXCEPTIONS_xx tables

The overall view on Harvester Exception (1 to 64) tables will be recreated to include the new column(s) added to these tables.

FUJ00090946

Synonym Definitions

i. Public Synonyms on TPS/SAV_PROD_MODE_SUMMARIES_xx views

There will be a new set of public synonyms created for the views on the current and saved set of Initial Summary tables. These public synonyms will be used by the TPS batch and support users (other than ops$tps) while accessing the Initial Summary tables.

The public synonym names and definitions will be similar to those for TPS Transaction tables.

ii. Public Synonyms on TPS/SAV_POL_FS_SUMMARIES_xx views

There will be a new set of public synonyms created for the views on the current and saved set of POL FS summaries tables. These public synonyms will be used by the TPS batch and support users (other than ops$tps) while accessing the POL FS summaries tables.

The public synonym names and definitions will be similar to those for TPS Transaction tables.

iii. Public Synonyms on Overall TPS/SAV_PROD_MODE_SUMMARIES_xx and TPS/SAV_POL_FS_SUMMARIES_xx views

There will be four public synonyms created for the overall views on the current and saved sets of Initial and POL FS summaries tables.

iv. Public Synonyms on Reference Data tables

There will be public synonyms created for the following new reference data tables:

i. TPS_POL_FS_ARTICLES
ii. TPS_POL_FS_ACCOUNTS
iii. TPS_POL_FS_MAPPINGS
iv. TPS_POL_FS_MAPPINGS_AT_S60
v. TPS_TRANS_MODE_CONVERSIONS
vi. TPS_POL_FS_EXCLUDED_MODES
vii. TPS_OPENFIG_CAC_PROD_MAPPINGS

v. Public Synonyms on other data tables

There will be public synonyms created for the following new data tables:

i. TPS_PROD_MODE_SUMMARIES_65
ii. TPS_POL_FS_SUMMARIES_INCOMP

FUJ00090946

Appendix C - User and Role Definitions

Role Definition

i. BSU, MONITORS and TPS_BATCH Roles

These are the existing roles in TPS database. The roles will be given object privileges on the following TPS objects as per the list below:

Views

TPS_PROD_MODE_SUMMARIES_n

SAV_ PROD_MODE_SUMMARIES _n

TPS_POL_FS_SUMMARIES_n

SAV_POL_FS_SUMMARIES_n

Where 'n' is the partition number ranging from 1 to 64.

Role Name	Privileges On Above Objects
BSU	SELECT
MONITORS	SELECT
TPS_BATCH	SELECT, UPDATE, INSERT, DELETE

Overall views

TPS_PROD_MODE_SUMMARIES

SAV_ PROD_MODE_SUMMARIES

TPS_POL_FS_SUMMARIES

SAV_POL_FS_SUMMARIES

Role Name	Privileges On Above Objects
MONITORS	SELECT
TPS_BATCH	SELECT

Transaction/Reference Data Tables

TPS_PROD_MODE_SUMMARIES_65

TPS_POL_FS_SUMMARIES_INCOMP

TPS_POL_FS_ARTICLES

TPS_POL_FS_ACCOUNTS

TPS_POL_FS_MAPPINGS

TPS_POL_FS_MAPPINGS_AT_S60

TPS_TRANS_MODE_CONVERSIONS

TPS_POL_FS_SUB_FILE_REGISTER

TPS_POL_FS_EXCLUDED_MODES

TPS_OPENFIG_CAC_PROD_MAPPINGS

FUJ00090946

Role Name	Privileges On Above Objects
BSU	SELECT
MONITORS	SELECT
TPS_BATCH	SELECT, UPDATE, INSERT, DELETE

FUJ00090946

Appendix D - POL FS Incomplete Summaries Report

```
FUJITSU SERVICES TRANSACTION PROCESSING SYSTEM                                                    Page No:    1
POL FS INCOMPLETE SUMMARIES REPORT FOR 26-MAR-2001*26
(LIST OF HELD UP POL FS SUMMARIES WHERE THE TOTAL VALUE FOR BRANCH AND TRADING DATE DID NOT BALANCE TO ZERO)
Report No.: TPSC<nnn>
Produced on: 02/04/2004  15:17:55
```

Trading Date	Group Id	Article Id	Article Type	Trans Mode Acronym	Settlement Type	Ledger	Account	Movement Type	Account Value	Account Quantity
05-Aug-2004	123456	1234567890	XX	ROAD	P	G	1234567890	123	S9999999.99	S999999999999999
05-Aug-2004	123456	1234567890	XX	RIAD	C	C	1234567890	123	S9999999.99	S999999999999999
05-Aug-2004	123456	1234567890	XX	DP	A	A	1234567890	123	S9999999.99	S999999999999999

Account Reference Id	Client Reference
123456789012345678901 2	1234567890123456
123456789012345678901 2	1234567890123456
123456789012345678901 2	1234567890123456

```
Total Number of Held Back Sub-file(s) = 1

*This is delta position for the date, not the cumulative position of POL FS Incomplete Summaries.

*** End of Report ***

Notes:
1. All numeric values should be right justified and character/alphanumeric values should be left justified.
2. Signs should be explicitly displayed for Amount and Quantity fields; '+' for positive and '-' for negative values.
```

[26] This is the logical processing date of the TPS System, i.e., the logical day the records would have been sent to POL FS if the total had balanced to zero.

FUJ00090946

Appendix E - Data for TPS_POL_FS_MAPPINGS_AT_S60 Table

The following spreadsheet contains locally defined mappings for pre-migration POL FS Summaries and the Balancing Transactions:

Pre_migration_pol_fs_mappings.xls

FUJ00090946

Appendix F - Data for TPS_OPENFIG_CAC_PROD_MAPPINGS Table

The following spreadsheet contains locally defined mappings for the cash account line to product mappings required to produce the Opening Figures:

"CAC_Mappings for Migration.xls"