FUJ00085124

Document Title: IMPACT Release 3 Counter Design for Balancing, Rollover and Stock Processing

Document Type: *High Level Design*

Release: *S80*

Abstract: This document describes the high-level design for the Horizon Counter Application Accounting Model and its Implementation for IMPACT Release 3.

Document Status: APPROVED

Originator & Dept: *Roger Donato, SI*

Contributors: *SI: Chris Bailey, Duncan McDonald, Walter Wright, Martin McConnell, Phil Orton, Peter Jobson, Roger York, Martin Nixon*
ASS: Gareth Jenkins

Internal Distribution:

External Distribution:

Approval Authorities:

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

FUJ00085124

0.0 Document Control

0.1 Document History

Version No.	Date	Reason for Issue	Associated CP/PinICL
0.1	04/06/04	First Draft	
0.2	22/06/04	Second Draft, incorporates informal review comments from Gareth Jenkins, makes minor refinements to the content and layout of version 0.1 as a result of continuous improvement. Introduces the second set of design changes; Introduction of Aggregation Engine for Volume Stock (5.1.3) and Appendix E – EPOSS Accounting Node Structure	
0.3	02/07/04	Third Draft. Introduces the third set of design changes Introduction of Trading Period Rollovers (5.1.2). Informal review comments on version 0.2, from Peter Jobson, Gareth Jenkins and Martin McConnell, have not yet been incorporated in this version	
0.4	29/07/04	Fourth draft. Introduces the forth set of design changes Amendment of Transaction Attributes for Volume Stock (5.1.4) Informal review comments on version 0.2, from Martin McConnell, Pete Jobson and Gareth Jenkins have been incorporated in this version Informal review comments on version 0.3 have been incorporated in this version The planned delivery structure of the HLD has been changed so that the content of increment 1 is enlarged as per Structure of Document (2.1) Inclusion of design content to resolve a number of gaps has been made, namely Application Message Expiry (5.1.1.2), "Unnecessary Message" Removal (5.1.1.3), Protection against Loss of Data (5.1.1.4), Improving Data Server Performance (5.1.3.3.2) Incorporation of a Changes Forecast which specifically addresses possible changes to content in previous versions of issued drafts	
0.5	25/08/04	Informal review comments on version 0.4 have been incorporated in this version Addition of a changes forecast considering reformat of document to eliminate excess content Fifth draft. Introduces the final sets of design changes contributing to increment 1 of the document:	

FUJ00085124

		• Introduction of Volume Stock Rollover Data Model (5.1.5) • Introduction of Trading Period Rollover Script (5.1.6) • Introduction of Volume Stock Balancing Model (5.1.7) • Introduction of Stock Balance Reports (5.1.8) • Introduction of Branch Trading Statement (5.1.9)	
0.6	14/10/04	Sixth Draft. Introduces the final sets of design changes contributing to increment 2 of the document: • Changes to Balancing impacted as a result of changes to Declarations, Discrepancies and Transaction Corrections (5.1.10) • Changes to Report Reprints (5.1.11) • Changes to Suspense Processing Additionally this version of the document adopts the design solution for implementation of the BTS, based on reference data item derivations This version of the document also incorporates some minor changes following discussions.	
0.7	09/12/04	Seventh Draft. Addresses all comments received in document versions to date. Completes detail in all areas of design.	
1.0	23/12/04	First Baseline. Addresses all comments received against document version 0.7 issued for formal review. The following changes have been made: • The format of start and end dates in the reference data collection specified in section 12.1 now includes a time element. The same changes has been reflected in RD/DES/056 • Correction of Mandatory Review Authorities, CS Service Introduction Manager being removed, as is already correctly an optional reviewer • Inclusion of design content for CP3842, pertaining to removal of Training Mode, allowing up to 99 Trading Periods and aligning giro reports to week numbers • Clarification of accounting sense on change to cash in pouches now using pairing product rather than reversing 5610 and correct of processing so that cash in pouches	

FUJ00085124

		• contra product identified by counter config params • Application of corrections resulting from comments from Mik Peach and Lina Kiang • Clarification of points relating to MiMan changes concerning withdrawal of chosen Accounting Period override, following comments received from Neil Gormley	
1.1	30/08/05	Addresses review comments received on version 1.0 from Gareth Jenkins. Incorporates changes for listed CPs and PEAKs. See section 0.5 for change details	CP 3888, 4002; PEAK 117079, 117170, 117784, 118648
2.0	12/09/05	Second Baseline	

0.2 Review Details

Review Comments by :	
Review Comments to :	*Originator*

Mandatory Review Authority	Name
ASS Designer	Gareth Jenkins
DU Development Manager	Andy Kennedy
CS System Support Centre Manager	Mik Peach
CS Network Service Manager	Peter Thompson
CS Data Centre & Operations Service Manager	Peter Thompson
CS Security Manager	Brian Pinder
Optional Review / Issued for Information	
DU Development Team	Mark Scardifield, Martin McConnell, Phil Orton, Jon Hulme, Rob Dinnadge
DU Test Design	Peter Robinson
CS Infrastructure & Availability Manager	Carl Marx
CS Service Introduction Manager	Graham Welsh
CS Service Definition Manager	Jane Collins
CS Major Release Manager	Peter Goodwin
CS Release Manager	John Budworth
ASS Technical Designer	Gareth Jenkins
CS Business Services Manager	Peter Thompson

(*) = Reviewers that returned comments

0.3 Associated Documents

FUJ00085124

Reference	Document	Version	Date	Title	Source
[TEMP]	PA/TEM/001			Fujitsu Services Document Template	PVCS
[DCRHLD]	EA/HLD/006			Impact Release 3 – Counter Design for Declaration, Correction and Revaluation	PVCS
[PROC]	DE/PRO/003			Post Office Account System Integration Lifecycle Processes	PVCS
[DP]	EA/DPR/004			IMPACT Release 3 Design Proposal	PVCS
[CDBT]	EA/CDE/002	1.0		Branch Trading Reporting, Management and Control and Transaction Management Conceptual Design	POL/POA via PVCS
[RDMCHLD]	RD/DES/056			Reference Data End to End High Level Design for S80 (Impact, Track & Trace, +1 Sales)	PVCS
[RPUI]	EA/IFS/011			IMPACT Release 3 - Report Production User Interface	PVCS
[DCRUI]	EA/IFS/012			IMPACT Release 3 - Declaration, Correction and Revaluation User Interface	PVCS
[BTSUI]	EA/IFS/013			IMPACT Release 3 - Balancing and Trading Statement User Interface	PVCS
[REPREC]	SD/DES/005			Horizon OPS Reports and Receipts - Post Office Account Horizon Office Platform Service	PVCS
[MENU]	SD/SPE/016			Horizon OPS Menu Hierarchy Release 2	PVCS
[MENU2]	SD/SPE/022			Horizon OPS Menu Hierarchy: Changes	PVCS

FUJ00085124

				Supplement	
[STYLE]	SD/STD/001			Horizon Office Platform Service Style Guide	PVCS
[MIGHLD]	EA/HLD/008			IMPACT Release 3 Migration High Level Design	PVCS
[RIPCNT]	TD/SPE/010			Riposte 6 Message Server Configuration for Counters	PVCS
[SOFTLAUNCH]	NB/LLD/056			SoftLaunch Low Level Design	PVCS
[RDDOC2]	RD/DOC/002			Reference Data Collections	PVCS
[RDREQ]	RD/DES/059			S80 Impact - Reference Data Requirements to support Functional Changes and to control Migration	PVCS

Unless a specific version is referred to above, reference should be made to the current approved versions of the documents.

0.4 Abbreviations/Definitions

Abbreviation	Definition
AG	Attribute Grammar
API	Application Programming Interface.
BP	Balance Period
BTS	Branch Trading Statement
CAP	Cash Account Period
DLL	Dynamic Link Library. A unit of executable code
DP	Design Proposal
EOD	End Of Day
EPOSS	Sub-system of Horizon desktop which is responsible for representing sales of goods
HLD	High-level design specification. See [PROC]
LLD	Low-level design specification. See [PROC]
MiECCO	Migration (ECCO). A tool developed to support the original migration to Horizon from the prior ECCO system (at some branches). No longer used.
MiMAN	Migration (Manual). A tool originally developed to support migration to

FUJ00085124

	Horizon from prior manual systems (at most branches), but also used to introduce new or temporary branches.
NT	Microsoft operating system
OBC	Operational Business Change. A mechanism for POL to request day to day changes to data within the Horizon system
POA	Post Office Account
POL	Post Office Ltd.
PPD	Procedures and Processes Document. See [PROC]
PVCS	Configuration Management tool used by Fujitsu Services (Post Office Account)
RAD	Rapid Application Development
RDMC	Reference Data Management Centre
RDT	Reference Data Team
RMS	Riposte Message Store
SLA	Service Level Agreement. An agreement, usually encapsulated in a contract, specifying the ways in which the delivery of a service will be measured and the level of such measures that must be achieved
SQL	Structured Query Language. This term is normally used in reference to the well-defined query languages used with relational databases. However, in this document, it refers to the proprietary Riposte query language syntax by which an application can specify which messages are to be retrieved from the Riposte message store. (It is thus more like the WHERE clause of a normal SQL SELECT).
TIS	Technical Interface Specification. See [PROC]
TP	Trading Period
TPS	Transaction Processing System; application which collects transaction information and passes it to various Post Office Ltd systems
UI	User Interface. The screen
VB	Visual Basic – development tool produced by Microsoft
VB6	VB version 6
VSS	Visual Source Safe

0.5 Changes in this Version

Version	Changes
0.1	None, this is the first version.
0.2	Comments incorporated as a result of Informal Review of first draft. Further, minor refinements to document content as a result of document reading

FUJ00085124

	Incorporation of design for Introduction of Aggregation Engine for Volume Stock (5.1.3)
0.3	Incorporation of design for Introduction of Trading Period Rollovers (5.1.2)
0.4	Comments incorporated from informal review of issues 0.2 and 0.3. Changes specified to planned delivery content of the HLD making up Increment 1 Included content resolving gaps in design, namely: • End of Day Protection of Data Loss • Start of Day Data Protection Loss • Data Server Performance Improvements • Results of analysis of message expiry from Gareth Jenkins and Jon Hulme Incorporation of design for Amendment of Transaction Attributes for Volume Stock (5.1.4)
0.5	This draft issue constitutes the planned content comprising increment 1 of the document. Comments have been incorporated from informal review of issue 0.4 Incorporation of design for Introduction of Volume Stock Rollover Data Model (5.1.5), Introduction of Trading Period Rollover Script (5.1.6), Introduction of Volume Stock Balancing Model (5.1.7), Introduction of Stock Balance Reports (5.1.8) and Introduction of Branch Trading Statement (5.1.9) have also been included.
0.6	This draft issue constitutes the planned content comprising increment 2 of the document, but also includes the draft content of the planned increment 3
0.7	This draft issue constitutes the planned deliverable constituting the full design for balancing, rollover and stock processing for formal review
1.0	First Baseline
1.1	Changes in response to review comments from Gareth Jenkins. Various editorial and functional clarifications/corrections : 5.1.2 clarifies rollover from Virtual TPs into the next year, determination of first TP, CAP/TP ambiguity, and simplifies the description of MiMAN 5.1.4 clarifies the timing of VolS and Tertiary Mapping changes. 5.1.3, 5.1.7 describe additional RCNTM/RCETM accumulators 5.1.8 clarifies Volume/Value stock reporting and use of RCNTM/RCETM accumulators 5.1.9 clarifies the derivation of the various elements of the BTS report, and delegates the actual layout definition to [BTSUI] 5.1.11 updated to remove Variance report (not in scope of this document) and clarify availability of reprints for Stock Unit Balance reports for BP rollovers

FUJ00085124

	5.1.12 clarifies the changes to the Suspense report from Point 20 onward 5.1.14 clarifies the implementation of client week numbers in response to CP3888. 9.1 contained obsolete material, and now delegates the softlaunch migration details to [MIGHLD] 11.0 corrects some omissions in the design proposal cross reference matrix
2.0	Originator changed to Roger Donato *(The original author - Phil Hemingway, and the person who applied the last updates to the document - Martin Nixon, have both left the Horizon project)* No technical changes since version 1.1

0.6 Changes Expected

Changes
The adoption of an enhanced version of Data Server is proposed for the production of all balance reports, providing a common solution to the requirement for processing stock by volume. However concern remains that the additional workload particularly for the Office Snapshot report will result in a notable degradation. A proposed architectural change has been put forward which promotes the use of checkpoint data to limit the amount of raw transaction data that needs to be read to produce any one balance report. At the very worst if there are performance issues with the office snapshot the BESReports component will have to be amended to adopt the same policy as has been introduced for DataServer Accumulators. It is therefore a subject of consideration to observe the performance of Data Aggregation for Impact Release 3 as to whether these forecast changes are implemented.

FUJ00085124

0.7 Table of Contents

FUJ00085124

FUJ00085124

1.0 Introduction

This document forms part of the overall Counter High Level Design for the implementation of Impact Release 3. The high level designs provide the counter design definition in response to the IMPACT Release 3 Design Proposal provided in [DP], which provides the solution proposal for the implementation of requirements stated in the Branch Trading Reporting, Management and Control and Transaction Management Conceptual Design, [CDBT]

This document is one of three design documents providing the counter design solution. It is complemented by the HLD for the Counter Declaration, Correction and Revaluation Functions of Impact Release 3, [DCRHLD]and the Migration High Level Design for Impact Release 3, [MIGHLD]with which this document should be read.

The document should also be read in association with Counter User Interface Definitions provided in the Impact Release 3 User Interfaces for Balancing and Trading Statements ([BTSUI]), Declaration, Correction and Revaluation ([DCRUI]) and Report Production ([RPUI]).

This document specifically provides a statement of the Counter Balancing, Rollover and Stock functions to be implemented as part of Impact Release 3. In this respect the document defines the overall balancing process initiated to carry forward figures to the next accounting period at stock and office levels, in response to transactions having been posted. The document therefore provides a model of the process involved and the implementation of that process. The design is less concerned with the actual dialogues the user undertakes to carry out the processes involved, which are little changed, and are specified in the above User Interface Specifications. In this respect the design reflects 'under the bonnet' aspects of the process. Any additional dialogues or changes to existing dialogues, which assist in an understanding of the entire process under change, or require specification, are included in this document.

The primary functional changes to the balancing process can be summarised as follows:

- The introduction of a 4 or 5 week Trading Period to replace the current one week Cash Account Period, as the mandatory balancing Accounting Period. Balance Periods are however retained.
- The elimination of the majority of stock handling by Value, to be replaced by the handling of stock by Volume. Methods of Payment and some specific existing Value Stock Items continue to follow the current processes.
- The removal of the Cash Account Report (in line with elimination of CAPs) and its replacement by a Branch Trading Statement.
- A revised Balancing Model where Receipts equals Payments (inline with the handling of Stock by Volume) in order that the branch can continues to reconcile its figures from one Accounting Period to the next.
- The removal of the distinction between Non-Value Stock and Value Stock, consequential to the management of stock by volume.
- The transfer of discrepancies to a local office suspense discrepancy.

All Changes to the balancing process can be attributed to the above primary process changes and it is possible to designate any change to the existing Counter Service as being attributable

FUJ00085124

to a change driven by the introduction of Monthly Trading Periods and/or the introduction of Stock Handling by Volume.

The design activity for the introduction of Impact Release 3 changes to the Counter is summarised in the following diagram.

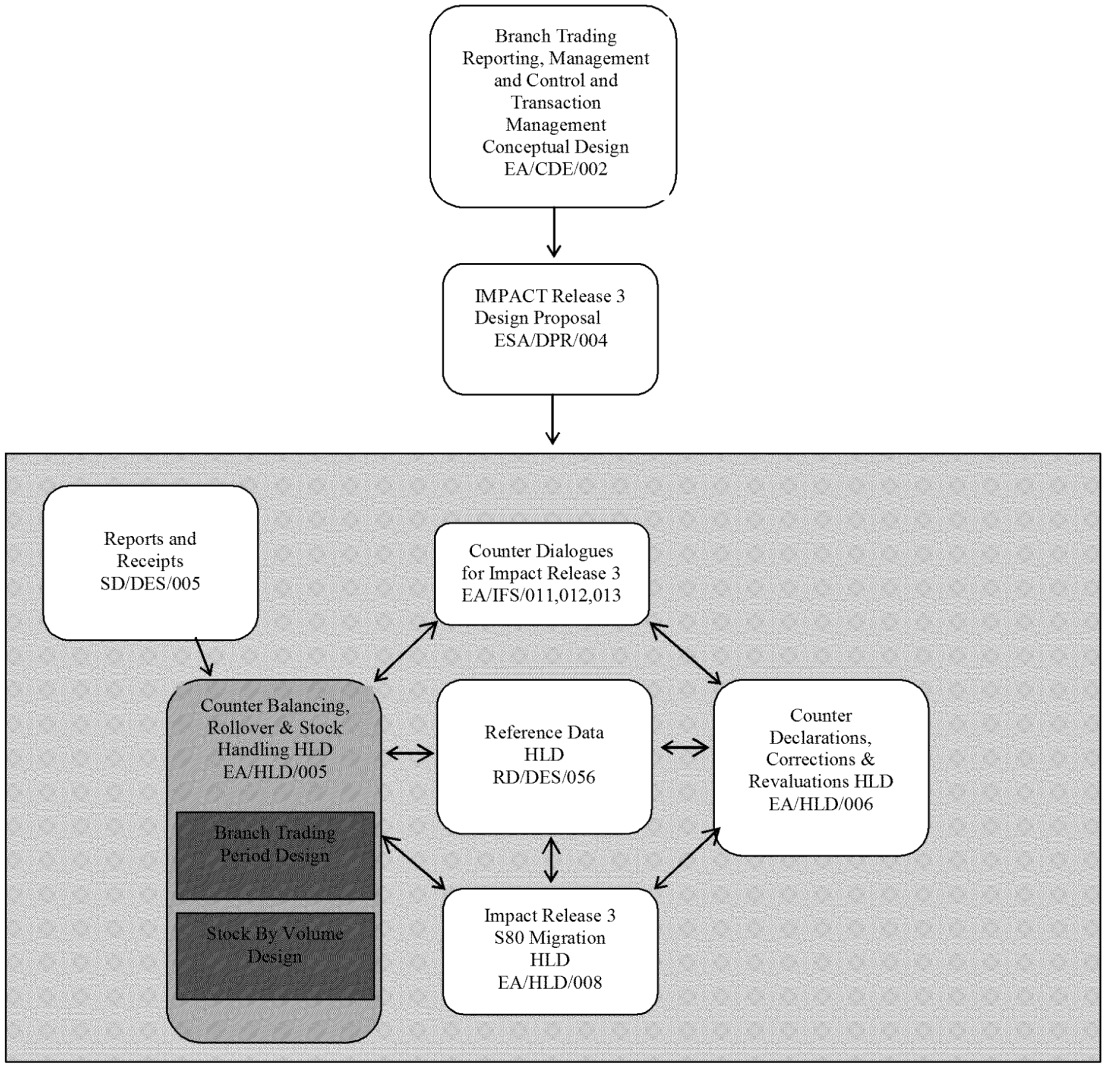


Figure 1 - S80 IMPACT Release 3 Counter Design Activities

FUJ00085124

2.0 Scope

This document forms part of the high level design for the IMPACT Release 3 Branch Trading project. The overall Post Office requirements for this project are stated in the 'Branch Trading Reporting, Management and Control and Transaction Management Conceptual Design' [CDBT], and the Design Proposal produced in response to these is the 'IMPACT Release 3 Design Proposal' [DP].

The document is only concerned with the design changes required to implement Impact Release 3 at the Counter, other changes being introduced to the data centre, and to the counter at S80 which are outside the scope of Impact, being also outside the scope of this document.

The proposal is supported by a set of High Level Design documents describing changes required at the data centre and branches. This particular document describes the changes required to the branch counters to handle 'Balancing, Rollover and Stock Processing', and is complemented by the high level design for 'Counter Declaration, Correction and Revaluation' [DCRHLD].

The scope of the changes required for Impact Release 3 have been analysed and it has been found that a natural split can be achieved in the changes required. This split reflects the nature of business processes involved in the operational day and at the end of the accounting period. At the start of day cash declarations may be undertaken. The following inday transactions are posted and taken at the end of the accounting period, for balancing, preceded by the declaration processing. This HLD is restricted to the changes to the counter to post inday transactions and undertake balancing.

There is some overlap with the high level design for the Counter Declaration, Correction and Revaluation Functions of Impact Release 3. This overlap will be stated where necessary but for the most part is limited to the accepting of discrepancies from declarations, which adjust the cash and stock value figures, which are then used in the balancing process.

Appendix A – Design Proposal Cross Reference Matrix defines the specific responsibilities for satisfying the counter aspects of [DP] between the two counter HLDs.

Part of that coordination is the entire subject of migration. The two HLDs both include elements of the Migration process to introduce Impact Release 3 adding further detail to subjects covered in the Migration HLD [MIGHLD].

Impact Release 3 Design Proposal identified a number of areas of change required to achieve the business proposition identified in the Conceptual Design. Not all changes defined impact the counter however numerous changes are interdependent with changes in other element of the system. For example Counter changes cannot be exercised without the requisite reference data being sent to the counter.

This design only defines the implementation of changes to the counter for S80. Those changes must be coordinated with other system changes.

The Counter balancing functions are implemented by the EPOSS Counter Application. The changes defined by the design principally require changes to EPOSS, but are not solely restricted to changes to EPOSS.

FUJ00085124

One anomaly in the inclusion of design changes as part of this document is the handling of stock by volume, which itself affects the process to remit and transfer stock. In this context the document includes transaction processing.

A second anomaly in the design is that stock by volume affects declarations. In this context there will be some overlap between the two HLDs.

The design focuses on the underlying processing of balancing, rollover and stock handling. It is not a focus of the document to specify the dialogues of balancing. However there must be an interaction between the dialogues document and the HLD. As an example the dialogues document will specify the dialogues and hence any messages involved. However the HLD will only describe the dialogues and will only relay that there are certain messages not what they are.

The objectives of this document are:

- To define a formal high level design in accordance with the Post Office Account Engineering process, that provides the design of Counter Balancing, Rollover and Stock Handling for Impact Release 3 at S80.
- To approach the design in such a way as to minimise risks and to acknowledge the timescale for the delivery of S80 in such a way that allows development to commence work on units of the design, before the complete design is finally approved. It is the intention that such a development will therefore employ a RAD approach.
- To provide a design that in being a delta to the existing Counter Service, describes the system as it currently exists so that the consequences of the changes can be fully appreciated, and to enable the strategic re-architecting programme within Post Office Account to make use of the functional elements of balancing within the Counter.

2.1 Structure of Document

Consideration to the design has addressed these objectives from the outset of document production and a structure for the document conceived to ensure all objectives are supported.

The changes to the counter for Balancing, Rollover and Stock Handling are defined in terms of the two principle business process changes, namely Use of Branch Trading Periods, and Stock Handling by Volume.

The project is not progressed according to the normal POA development methodology as described in [PROC] – i.e. via production and review of the conceptual design (requirement statement), design proposal (specification), high level design and low level design documents before proceeding to implementation and validation. Whilst the conceptual design and design proposal have been produced, the process of HLD, LLD and development will be undertaken though joint working and RAD.

However, the disparate nature of the anticipated changes to existing design and implementation, together with the required tight development schedule, mean that for this particular design a more aggressive approach is required :-

The first difference is that instead of concluding formalities for the entire high level design phase before low level design can begin, the high level design is to be broken down into independent design changes (as far as is practical), so that low level design can proceed on particular changes without waiting for other independent changes to be completed.

FUJ00085124

There is nothing new in this – it is quite normal for a Design Proposal to be addressed by a set of independent High Level Designs, and for each of these to be progressed in parallel. The change here is that this particular High Level Design document is itself to be regarded as covering a set of independent functional units, such that particular units can be progressed to Low Level Design and implementation before other units are necessarily firmed up.

The second difference occurs because the functional units require (indeed must require because of timescales) discrete changes to existing low level design and implementation. The best way to minimize the elapsed time expended on these changes is to take advantage of their small size and to iterate them by focussed technical workshops involving people familiar with the existing high and low level designs. This RAD approach effectively allows the low level design changes to proceed in parallel with the high level design. (Indeed, it may be equally practical for some of the low level design issues to be explored by proceeding with implementation in parallel – at least in prototype. Further consideration of that option is outside the scope of this document).

The high level design covered by this document is therefore broken down into a number of relatively independent functional units. An agreed design production plan has been considered and compiled which structures the design of the functional units into a number of cohesive sets, each set forming a document increment.

The design of each functional unit will be issued as a draft version of the document, for informal review. Successive draft revisions of the document will add details for each of the units, whilst also reworking informal review comments to previous draft versions.

Following the agreed structure draft versions of the document will be issued for functional units until all units forming the group assigned to an increment have been addressed, when the document in that form will be issued for formal review.

Thus increment 1, for example, will describe functional units A, B and C say to a level suitable for formal review and acceptance. Increment 2 will then include functional units D and E to reviewable status, and Increment 3 will finally describe F. Review of increments 1 and 2 should clearly be focussed on units A, B, C and D, E respectively. Any feedback on other units is of course welcome, but may not be incorporated until the increment in which they are due for review.

In an ideal world, review of new material for (say) unit D would not impact the design for units A, B, C. but it is possible that there will be limited rework required. Because of this, review of the final increment must cover all the units.

The following functional units and document increments have been identified.

Increment	Primary Business Process	Function Change
1	Introduction of Trading Periods	Extension of Transaction Retention (5.1.1)
		Introduction of Trading Period Rollovers (5.1.2)
	Handle Stock by Volume	Introduction of Aggregation Engine for Volume Stock (5.1.3)
		Amendment of Transaction Attributes for Volume Stock (5.1.4)
		Introduction of Volume Stock Rollover Data Model (5.1.5)
		Introduction of Trading Period Rollover Script (5.1.6)

FUJ00085124

		Introduction of Volume Stock Balancing Model (5.1.7)
		Introduction of Stock Balance Reports (5.1.8)
		Introduction of Branch Trading Statement (5.1.9)
2	Handle Stock by Volume	Adoption of Variance Handling from Discrepancies (5.1.10)
		Introduction of Extended Reprints (5.1.11)
		Changes to Suspense Account (5.1.12)
3	Introduction of Trading Periods	Adoption of Cut Off Reports for Trading Periods (5.1.13)
		Amendment/Removal of Generic Reports (5.1.14)

Table 1 - Functional Units and Document Increments

The substance of this document comprises a number of delta changes. These define the design of individual functional changes. Each functional change provides a design definition to achieve a specific business change, contributing to the whole. Their purpose is primarily to inform the low-level design and development process, a primary purpose of this document.

FUJ00085124

3.0 Design Principles

3.1 Incremental Design

The design of the changes for Impact Release 3 is adopting an incremental approach. That is that the overall solution will be built by a set of interdependent changes each achieving a functional enhancement. Such an approach is considered as one of the ways to ensure the risks identified in Section 10 are mitigated.

3.2 Joint Working

The incremental design will be constructed using a Joint Working activity with Development. The joint working activity will use the opportunity to propose, agree and prototype design changes earlier than otherwise would be the case, providing assurance that the complete solution can be achieved within the timescales and also minimising the risks identified in Section 10.

The joint working activity is performed in a controlled fashion.

3.3 Design Neutrality

The design of this system should wherever possible, adopt a neutral approach. That is where the behaviour of the system can be related to some other aspect, it should derive its behaviour by reference, and not be hard-coded.

3.4 Data Driven functionality

The system is designed to be data driven. A clear demarcation between logic and data needs to be maintained. Logic is encapsulated in the code, data in the reference data. However, it is never that simple. Some paths through the logic depend on the data; this is unavoidable. What is undesirable is for this to go to the level where the logic becomes convoluted.

3.5 Reusability

This is closely related to Data Driven functionality. Where two components are sufficiently similar, the same should be used in both cases.

3.6 User Interface, Screens and Prints

All user interactions will conform to [STYLE] wherever possible, to provide the clerk with an image consistent with the rest of the system. Appropriate prompts will be provided to guide the clerk through the transactions.

In all dialogues with the user, any visual component must have an equivalent keystroke. Any keystroke that has an effect on the system will have a visual component that will have like effect.

References [RPUI], [DCRUI], and [BTSUI] describe the screen flows and [REPREC] the print layouts for receipts and reports.

FUJ00085124

3.7 Reference Data

All application reference data used by the system to provide 'parameter' data will be temporal and must therefore be accessed through appropriate interfaces that provide support for this.

The order of attributes will not be relied on in accessing reference data, unless there is specifically an ordering declared. Any attribute that is not mandatory may be omitted, so empty and missing attributes must be treated identically. The presence of other attributes must not cause failure. Any validation of attributes on reference data must be limited to checking the presence of mandatory attributes, and the values of those parameters present.

[RDMCHLD] provides the specification of Post Office Reference Data changes to support Impact Release 3 at the Counter. Appendix B – Affected Reference Data Collections within this document also specifies the Type C Reference Data providing Configuration Data to the Application changes.

[RDREQ] provides a specification of the Reference Data Requirements to support the functional changes and migration.

3.8 Persistent Objects

The application makes extensive use of persistent objects. Persistent Objects are internal message store objects generated and maintained by the counter applications. Their format in the message store is similar in nature to reference data however they can and are updated by the application, unlike reference data which is read only.

Persistent Objects form an important part of the application providing formal temporary storage of variables or attributes as an interface between different parts of the application. The design and introduction of Impact Release 3 will utilise and build on existing persistent object structures to achieve an efficient solution and promote concepts already in use.

The introduction of any new additional persistent objects will be considered carefully as part of design so as not to unduly impact counter operation, as their implementation carries a performance overhead in their indexes being maintained.

Similarly any persistent objects removed from use will not actually be deleted, as differences in the archiving and index timing between the counters and correspondence servers can lead to problems if the same objects are recreated at a later time, and particularly if within a year of their previous deletion.

FUJ00085124

4.0 Requirements

Requirements for Impact Release 3 are documented in [CDBT] and are cross-referenced in [DP].

FUJ00085124

5.0 System Components

The changes required to the Counter in response to the requirements for Impact Release 3 for Balancing, Rollover and Stock Handling are mainly, but not entirely, attributable to EPOSS. Each change to the system is defined in terms of a Functional change to EPOSS. Changes to other components where required are identified specifically.

5.1 Application Components

5.1.1 Extension of Transaction Retention

The lifespan of a message written to the riposte message store by counter applications is governed by the coordinated management of counter applications and Riposte. An overview is provided in Appendix D – Overview of Riposte Message Expiry.

Currently branch balancing is required each week, with occasional allowable extensions through the Extended CAP functionality. The move to a monthly trading period is in conflict with the current expiry periods assigned to transactions and other messages within the branch counter Riposte Message Store.

The impact is made worse by there being a minimal coordinated approach to the assignment of actual expiry periods to messages within counter applications.

Changes are required in a number of areas to extend transaction and message retention in order that messages are not deleted before their use in the balancing process has been satisfied.

- Changes to Default Riposte Counter Configuration Parameters.
- Changes to the Counter Applications to use new assigned expiry period mechanism.
- Removal of unnecessary message expiries.
- Protection against loss of data.

The changes required satisfy the proposals made in [DP] at 2.5.1.1.1.

5.1.1.1 Riposte Configuration Parameters

Riposte Configuration Parameters at the counter control the assignment of message expiry values to written messages in the counter message store, when no expiry value is provided by the application writing the message or when the expiry provided is inconsistent with a set of predefined configuration limits

The Riposte Configuration Parameters applicable to the Counter will change as follows:

Configuration Parameter	Old Value	New Value
MaxMessageExpiry	37	50
DefaultMessageExpiry	36	43
MaxReanimation	36	49

Table 2 - Riposte Configuration Parameter Settings

The Counter MinMessageExpiry value remains unchanged at 34.

FUJ00085124

Changes to Riposte Configuration Parameters on the Correspondence Server are outside the scope of this document.

Changes to the Riposte Configuration Parameters specified above will also be separately documented in [RIPCNT], the definitive documentation of the Riposte Configuration Parameters.

The changes required satisfy the proposals made in [DP] at 2.5.1.1.1.1.

5.1.1.2 Application Message Expiry

Individual counter applications are responsible for their own message expiry values, and within applications distinct functions are responsible for assigning the appropriate value to that function. In most cases expiry values are hard coded.

Analysis of the primary messages written in everyday use of the counter, by applications hosted on the counter reveals a number of different values to be used. As an example the specific value of 35 days is used, for the assignment of Message Expiry for EPOSS Transactions. It is clear that such an assignment is insufficient for balancing on a monthly basis, given that the actual balancing process may not be undertaken for a number of days after the monthly period has expired.

A change is required to adopt the assignment of the expiry period for EPOSS messages from Reference Data, hence allowing the value to be adjusted without impacting counter applications. This will affect any counter application writing transactions contributing to the Branch balancing process.

The following scheme shall be employed:

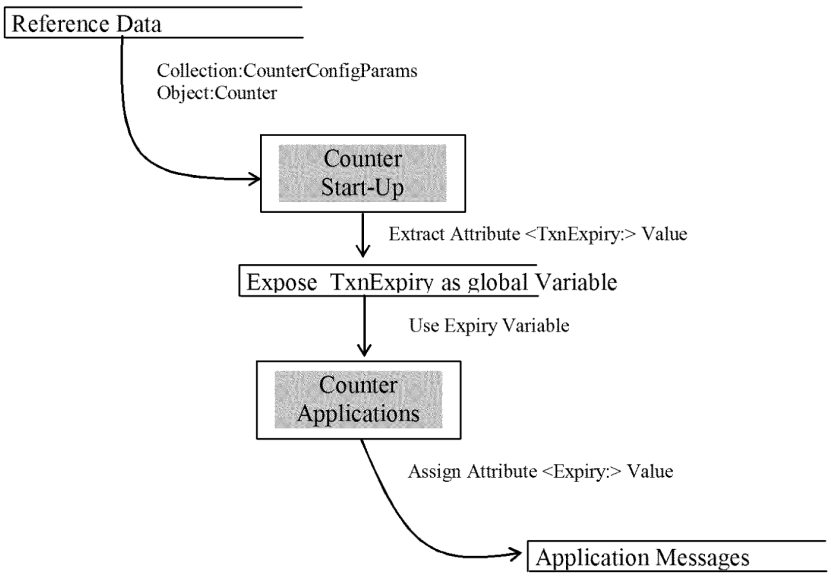


Figure 2 – Setting Message Expiry Periods from Reference Data

Counter Application Transaction Expiry Period Assignment will be based upon a value held in the new Type C Reference Data Collection <CounterConfigParams:>. An object may exist for

FUJ00085124

each Counter Application managing its own Message Expiry Periods. However a Counter Application default object will also exist. The collection is defined in 12.0.

Counter Applications defining their own Start-Up processing will read the appropriate collection and extract the <TxnExpiry:> attribute and expose the value to the application for the duration of the service life.

Counter Applications recording messages to the message store will assign the <Expiry:> attribute from the exposed variable.

The scheme affects Messages currently written with a hard coded expiry of 35 days and ideally should be applied to all messages which support Balancing. A reference data value of 42 days is initially proposed. The following table defines those message types which are needed for reporting throughout the longer duration of trading periods and so should have their expiry periods changed to use the scheme described.

Classification	Comment
Cash Account	Impacted but will be removed as part of other changes for Impact R3
Cut Off	
Declaration	
Event	
OpeningFigures	
Rollover	
Txn	Include APS Txns
Migration	Impacted but requires further consideration as to strategic Solution
Counter EOD	Used by Counter EOD on the Slave
EOD	EOD on the Gateway
EOD HT	EOD on the Gateway

The following components are impacted by the change:

EPOSSCore
EPOSSDeclare
EPOSSMessage
EPOSSStockUnit
ReportProcessor
LFSConfirm
OBCS
End of Day

The changes required satisfy the proposals made in [DP] at 2.5.1.1.1.2.

5.1.1.3 "Unnecessary Message" Removal

Analysis of message expiry assignment has revealed a less than coordinated approach to message expiry assignment. There are two risks opened up as a result:

- In adopting an approach that provides for the assignment of message expiry from reference data it is possible that a message currently assigned a constant expiry may be missed and a resultant message loss occurs.
- In lengthening the balancing period to a month it is possible that some messages will

FUJ00085124

now remain in the message store unnecessarily.

As a result a second attribute is defined in the Type C Reference Data Collection <CounterConfigParams:>, <MinAppExpiry:>. Use of the this attribute should be adopted in writing any message currently written with a hard coded value of 35 days, for those message types which are not required to support retention through the longer trading periods. As a result the actual expiry value in the Riposte message will be replaced by the Riposte Configuration Parameter MinMessageExpiry, overriding that assigned from MinAppExpiry. A reference data value of 20 days is initially proposed for MinAppExpiry.

Counter Applications recording messages to the message store will assign the <Expiry:> attribute from the attribute <MinAppExpiry:>.

An analysis of messages which should have their message expiry reduced or removed is as follows, along with the modules/DLLs affected.

Classification	Comment
NetworkQOS	Network Stats, but not written by Counter Applications and therefore no change required within the Counter Code
LPO	Used by APS

The following components are impacted by the change:

APS

There also exists a category of message, normally of low volume, for which the Expiry Period can be specifically assigned on a case by case basis, normally a very low value. In such cases either case specific reference data items can be defined or the message expiry can be hard coded. An analysis of the message types which should have their message Expiry specifically assigned is as follows, along with the modules/DLLs affected.

Classification	Comment
CtrListen	Msgs form a cross-counter locking mechanism
InterfaceLock	UI button locking mechanism
PrintInfo	
OBCS Admin	

The following components are impacted by this change:

BESReports
EPSOSCommon
EPOSSReportProcessor
EPOSSStockUnit
EPOSSWatchDog
MiMan
NBFinalise
NBRequestReply
OBCS
TestPeripherals

Currently the components impacted by this category of message will be left unchanged, the hard coded value being retained.

FUJ00085124

There remains a set of messages which are written which adopt the default expiry values specified in the Riposte Configuration parameters, and for which the application controlled expiry is either not specified or is of an arbitrary value (below the Riposte default). The following analysis shows the message types and what is the recommended value for the Expiry in these cases. However they will only be changed if the code is being changed in the area of Impact Release 3.

Classification	New Expiry	Comment
APS Recon	20	
DailyCACT	20	
DailyTxnCT	20	Will be removed
Elapsed	20	Generated by XSLT processing?
Escher	Default	We can't change Escher's code so these will stay as they are
Event	42	Event Ids 47 & 48. Should be explicitly set like other Events
KeyMsg	??	Leave since no changes needed to KMA
LFS	??	Need further analysis. Suggest they stay as Default
LFS Remout	42	
Mails Print	20	
Mails Txn	42	Some are EPOSSTransactions
Origin	20	Generated by Mails?
PAF	20	
Ping	1	
RIAD Txn	42	
SCCache	5	
SessionTransfer	5	
Test	Default	Will not occur on real counters
WeeklyEPOSS	20	Will be removed as part of Impact Release 3 anyway

The changes required satisfy the proposals made in [DP] at 2.5.1.1.1.3.

5.1.1.4 Protection against Loss of Data

Whilst the changes to message retention assignment described above are sufficient for branches operating in steady state there is still a risk to data loss and real examples where such loss can indeed occur. [DP] describes such examples.

The inclusion of processing to protect the system from loss of data is the responsibility of System Start Up and End Of Day.

5.1.1.4.1 System Start Up

The changes required satisfy the proposals made in [DP] at 2.5.1.1.1.4

The starting of the Riposte Service after a reboot is instigated from a batch file (services.bat) run under the control of PostPolo.exe launched by Polo. This batch file is one of a number in the c:\admincfg\PostPolo directory which are run in sequence by PostPolo.

FUJ00085124

Changes are required to insert additional batch files to run before services.bat which change the Riposte Configuration required if necessary so prohibiting Riposte from archiving old messages.

The changes required are different depending whether the reboot is a live counter rebooting after a closure or the implementation of a spare replacement.

Live Counter Reboot

A new batch file will be introduced in c:\admincfg\EachTime.

The batch file will instigate actions to examine the AdminRebootDetect service and determine how long the counter has been down. The AdminRebootDetect service records the times and types of system down events which can be used to establish whether Riposte Archiving needs to be switched off. If the counter has been switched off or down for more than 24 hours then the Riposte Configuration Registry settings, DisableArchiving, must be set to 1. Where the condition 'no data available' is met, for example as obtained by AdminRebootDetect from its analysis of evidence from the event log, the default action will be to disable archiving.

End of Day checks will be responsible for resetting the registry settings back to 0.

Box-swap Counter Reboot

The implementation of a spare will implement a slightly different approach.

The Cold Build will unconditionally switch off Riposte Archiving, setting the Riposte Configuration Registry setting, DisableArchiving, to 1. See [RIPCNT].

5.1.1.4.2 End of Day

The changes required to End of Day Processes are now described in [DCRHLD].

FUJ00085124

5.1.2 Introduction of Trading Period Rollovers

Impact Release 3 sees the replacement of the one week Cash Account Period by a four or five week Trading Period as the mandatory period against which branches are required to balance their accounts.

The introduction of trading periods carries a number of specific business changes, such as the introduction of the Branch Trading Statement to replace the Cash Account. However the introduction of Trading Periods can also be seen as a generic and largely mechanical change, to change the accounting schedule and merely replace the CAP by TP.

Note that unlike the existing Cash Account, the Branch Trading Statement will not be used by POL for accounting purposes. It is purely a Management Tool to help the Postmaster manage his / her branch and to prove to an auditor that they are in control. POL will now be doing all its accounting based on the transactions fed through each day to POL FS.

5.1.2.1 Scope

This change restricts its scope to the physical generic change of period from weekly CAP rollover to monthly TP rollover, and the impact of that change on legends in the user interface and printed reports.

Business functionality is being introduced that is associated with a TP rollover which is itself different to CAP rollover, but those changes are covered in separate sections. See Introduction of Volume Stock Rollover Data Model (5.1.5), Introduction of Trading Period Rollover Script (5.1.6), Introduction of Stock Balance Reports (5.1.8), Introduction of Branch Trading Statement (5.1.9), and Introduction of Extended Reprints (5.1.11).

5.1.2.2 Introduction to Accounting Periods

Accounting within Post Office branches is based around the concept of Periods. All Transaction Postings made are assigned to a given period. Reports produced are then populated from transactions for that period, transaction totals are summarised at the end of the period, and totals are carried forward as initial figures for the start of the next period. The following terms of period are relevant.

Accounting Year

This is the financial reporting year for Post Office Ltd. The mapping between accounting year and physical dates is defined by Reference Data.

It has no direct impact on accounting within the branches, except that accounting periods are contained within particular accounting years. There are no specific accounting year end processes within the branch.

Accounting Period

An accounting year is divided into a number of Accounting Periods, and the mapping between accounting periods and physical dates for a particular accounting year is defined by Reference Data. Hitherto it has had no direct impact on accounting within the branches, but with the introduction of Impact Release 3, it will form the basis for defining Trading Periods (see below).

An accounting period is typically a month long, but during the transition from Cash Account Periods to Trading Periods, the initial Accounting Periods (and hence Trading Periods) may be

FUJ00085124

reduced to as little as a week in order to complete pilot transitions in the shortest practical time.

Cash Account Period (CAP)

Hitherto, the Cash Account Period has underpinned accounting control within each branch. At the end of each CAP, production of a formal Cash Account report has been a prerequisite for allowing the branch to 'rollover' into the next CAP. The mapping between CAPs and physical dates for a particular accounting year is defined by Reference Data, and CAPs are contained within particular Accounting Periods.

A CAP is typically a week long – starting on Thursday and ending on Wednesday, but individual branches have the flexibility to 'extend' the CAP by up to 2 weeks to allow for staff absence when the Cash Account report is normally due. In addition to that, there is no absolute requirement to 'rollover' the CAP on Wednesday each week. Horizon issues warnings if the CAP is rolled early or late, but does not prevent it.

(Strictly speaking, it warns you at login if the date is earlier/later than the CAP for the Stock Unit. It also warns you at rollover if the date is earlier than the CAP end date. It does not warn you if the rollover is late (presumably on the grounds you've already been warned once at logon!)

Trading Period (TP)

With the introduction of Impact Release 3, branches no longer need to produce a weekly formal Cash Account report, because transaction analysis is reported centrally via POLFS on a daily basis. Because of that, the need for rollover between Cash Accounting Periods is removed, and branches instead produce a Branch Trading Statement at the end of each so-called Trading Period.

A Trading Period is based on an Accounting Period, with the start and end dates offset by an amount to smooth the central workload. The offset is defined by branch-specific reference data. Note that because of the branch-specific offset, CAPs are not bound by particular TPs. In other words, you can't deduce that CAP N will roll into TP M just because N is bounded by a particular Accounting Period.

The existing branch flexibility in being able to 'extend' a CAP by up to 2 weeks is not carried forward to TP (because allowing 2-3 months of unaccounted transactions is deemed unacceptable). Nevertheless, Horizon will not enforce strict TP rollover on a particular date. If branch staff choose to roll the TP early or late, Horizon will issue warnings (as now), but will not prevent it.

In addition, because of the risk of data loss arising from message expiry, a warning will be issued at logon if the Stock Unit has not been rolled over within the configured RollExpiry period (see Extension of Transaction Retention 5.1.1). Note that this condition can occur even when the current TP is appropriate for the current date, because the Stock Unit might have rolled into the TP early, and have old messages as a result.

Balancing Period (BP)

Within a branch, individual Stock Units (the mechanisms for asset control and cash balancing) can subdivide a CAP into a number of Balancing Periods in order to manage cash and assets more closely. This is useful in high throughput branches, and for shared Stock Units where a

FUJ00085124

number of staff share responsibility for the assets. Balancing Periods have no fixed start and end dates, but they are bounded by a particular CAP.

The use of balance Periods is carried forward as a feature, for use with TPs also.

Stock Unit and Office Rollovers

When a particular Stock Unit declares and commits its assets by producing a Balance report, it can 'rollover' to start a new period. This can either be the next BP within the current CAP/TP, or it can roll into a new CAP/TP. Once all Stock Units have rolled into the new CAP/TP, the Office can itself be rolled forward and appropriate Cash Account or Branch Trading Statement report produced.

During the time that a Stock Unit is 'ahead' of the Office, it can continue with normal business and can checkpoint its asset declarations by rolling into new BPs as desired. However, Stock Units can only be one CAP/TP ahead – in other words they cannot roll forward into the next CAP/TP until the Office has entered the current one.

Note that the identity of the new CAP/TP is not simply 'old+1'. Because of the ability to 'extend' the CAP by up to 2 weeks, Stock Units can roll from (say) CAP *N* to CAP *N+3*. In addition, at the end of the Accounting Year, the last CAP may be CAP 52 or CAP 53, and is followed by CAP 1 for the next year.

Determination of the next CAP takes no account of the current physical date – it is always by lookup from Reference Data, and is based on the current CAP.

For example, if branch staff ignored Horizon warnings and omitted to rollover a CAP until 6 weeks late, then on finally doing so, they would enter the next CAP (which itself would be now 5 weeks late for rollover) rather than the one appropriate to the current date. Having done so, they would have to repeat the rollover process until reports had been produced for each CAP. This is in order to provide all information necessary for POL to have complete figures for each CAP for each branch.

Avoidance of such a situation is maintained through managerial oversight from POL of its branch network activity.

The same will apply for TP rollovers, except that TPs cannot be 'extended'.

5.1.2.3 Approach Taken to Transition

The transition from CAP-based to TP-based accounting is required to be on a branch-by-branch and StockUnit-by-StockUnit basis. This means that at particular times during the transition of the estate and of individual branches the software and reference data must support concurrent use of both TP-based and CAP-based accounting, to allow for Stock Units that have or have not yet rolled forward from a CAP into a TP.

In order to minimize the risk associated with a large scale pervasive change, use of the term CAP will be retained throughout the code, existing reference data and persistent objects - the semantics will simply be changed to mean Trading Period rather than Cash Account Period.

At a chosen CAP for each branch (the so-called 'Final CAP'), transition will commence with the first Stock Unit rollover from that CAP into an associated first TP. The other Stock Units within branch subsequently roll into the TP, followed finally by rolling the Office. Thereafter the Stock Unit and Office are supported purely by TP-based accounting.

FUJ00085124

For ease of piloting, different CAPs are identified for different branches across the estate, and transition occurs over a period of time until all are complete.

For a given branch, once the transition has occurred, all occurrences of the term CAP in the User Interface or printed reports must be replaced by the term TP or Trading Period as appropriate. However, during the transitional period that a particular Stock Unit is operating in TP mode, but others are still operating in CAP mode, there is a need to be able to display either CAP or TP legends according to the Stock Unit to which the current user is attached.

5.1.2.4 Management of Steady-State CAP Rollover

The following describes the way in which Stock Units and Offices currently roll forward from one CAP into the next.

The existing EPOSS implementation uses two persistent object collections in Riposte message store to represent the branch and its Stock Units respectively. These describe the CAP and BP in which they are operating.

The 'EPOSSCAP' collection (see 13.1) contains an object 'Office' that describes (among other things), the CAP in which the Office (branch) is operating, the CAP from which it last rolled, and the CAP into which it will next roll.

The 'StockUnits' collection (see 13.2) contains an object (with name <stock unit id>) for each Stock Unit in the Office Each such object describes (among other things) the CAP and BP in which the Stock Unit is operating, and the CAP into which it will next roll.

Note that the 'current' and 'next' CAP values for a Stock Unit do not necessarily match those of the Office because a particular Stock Unit will be ahead of the Office once it has rolled over to the next CAP.

The new 'next CAP' following Stock Unit or Office rollover is determined from on the Post Office accounting calendar, represented by the 'CAWeek' temporal Reference Data collection in Riposte message store, as described in [RDREQ]. Each member of the collection represents one CAP in one particular accounting year, and the object name is of the form <year><cap>. (See Figure 3)

FUJ00085124

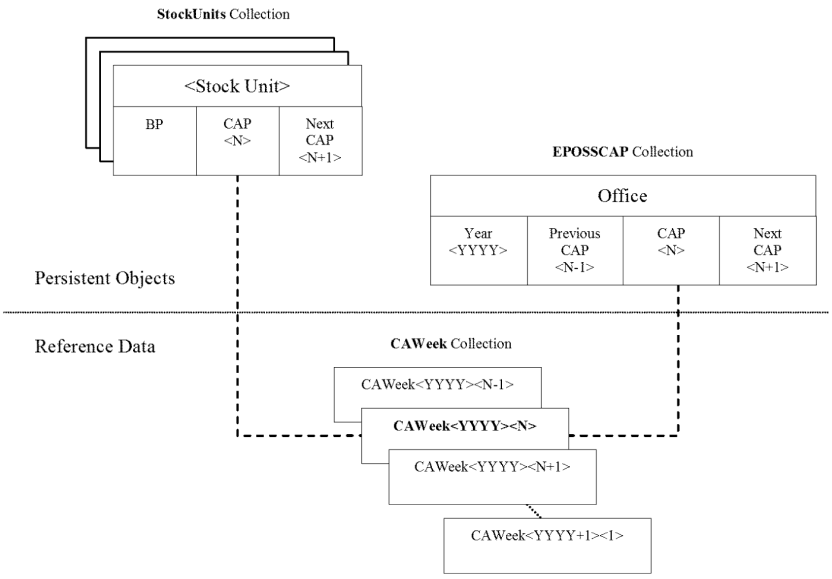


Figure 3 - CAPs and the CAWeek Accounting Calendar

When a Stock Unit rolls over from one BP to the next, the BP value is simply incremented by one. When it rolls over from one CAP to the next, the BP value is reset to 1, the current CAP value becomes the next, and a new 'next CAP' value is determined (see below).

Once all the Stock Units have rolled over to the next CAP, the Office itself can be rolled. As for the Stock Units, the previous CAP value for the Office becomes the current, the current CAP value becomes the next, and a new 'next CAP' value is determined (see below).

The algorithm to determine a new 'next CAP' value is that the existing 'next CAP' value is incremented by one, and that CAP is looked up in the Accounting Calendar for the year associated with the existing 'next CAP'. If the new CAP exists, then that is the value chosen. However, at year end that CAP may not be present (i.e. when incrementing CAP from 52 to 53 or perhaps from 53 to 54). In that case, the year is incremented by one, and CAP 1 is looked up in the calendar for the new year.

Rollover and selection of the next CAP is not affected by the actual physical date on which the action is undertaken. Nevertheless, as separate business rules, Horizon does warn about attempting early rollover, or logging in when the current CAP for the attached Stock Unit should already have been rolled. Both of these checks are based on start and end date properties for the CAP – obtained from the CAWeek accounting calendar 'CAWkSD' and 'CAWkED' attributes.

Hitherto, CAP Extension has been handled by adjusting the 'next CAP' value for the Office and all the Stock Units, and is not permitted if any of the Stock Units have already rolled into the original 'next CAP', or if the requested week would be in a different accounting year. However, this functionality will be removed from S80 (see [MIGHLD] and section 9.1).

FUJ00085124

5.1.2.5 Management of Steady-State TP Rollover

With the advent of TP rollover instead of CAP, the same logic applies unchanged, except that

1. CAP Extension is not supported for TPs. The extension logic can simply be removed, because the facility is to be withdrawn at S80 even for CAPs, in readiness for transition to TPs. See [MIGHLD] and section 9.1
2. The accounting calendar is represented by a different temporal Reference Data collection 'AccountingPeriods', as described in [RDMCHLD] and Appendix B – Affected Reference Data Collections. Here, each member of the collection represents a particular accounting year (the object name is of the form <year>), and contains attributes named <ap>– one for each Accounting Period, which corresponds 1:1 with a Trading Period. See Figure 4.
3. The algorithm for ascertaining the next TP is the same as described for determining the next CAP, because the year end condition can be detected in just the same way. However, if the proposed AP/TP does not exist in the current year, and the accounting calendar for the next year is not yet available (which may be a likely scenario in the event of branches rolling too frequently in the early life of Impact), then additional 'virtual' trading periods can be generated, a new requirement directed by CP 3842, by simply adding 1 to the current TP. In that situation, the software will select successive virtual TPs until the accounting calendar becomes available, or until TP99 is reached. In either case, the next TP selected will be TP1 of the next year.
4. Only when 99 periods have been used should the software assume TP 1 in the next financial year. The use of virtual trading periods should only be used in the absence of calendar data and may result in a rollover from a virtual Trading Period to a physical trading period that is not contiguous, when missing calendar data is delivered.

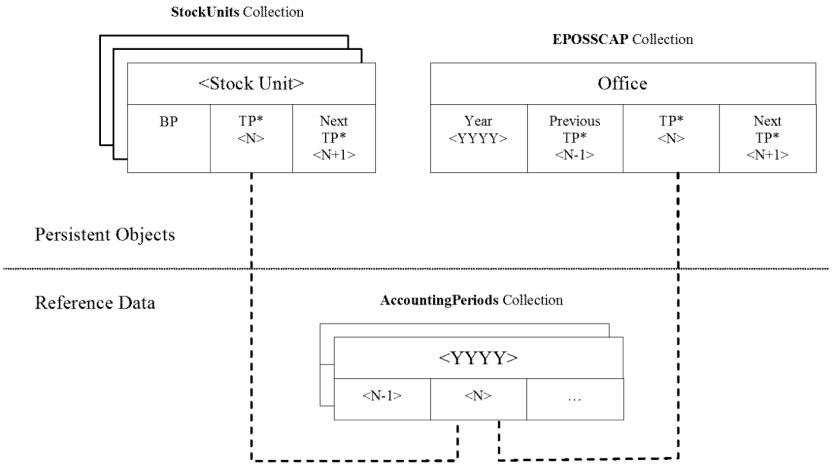


Figure 4 – TPs and the AccountingPeriods Calendar

FUJ00085124

5. As implied earlier, rollover to BPs is unaffected by the change from CAP to TP – the BP number is simply incremented by 1 each time.

6. The business rules to detect early or late rollover will be based on the starting and ending dates for the current TP (rather than CAP). Note that although the AccountingPeriods calendar defines the start/end dates for each Accounting Period, the actual Trading Period for a particular branch is offset from that to smooth the central POL workload at the end of each Accounting Period. That offset is held in days, as an attribute of the branch-specific OutletDetails object within the temporal Reference Data collection 'Outlet'. See Appendix B – Affected Reference Data Collections and Figure 5

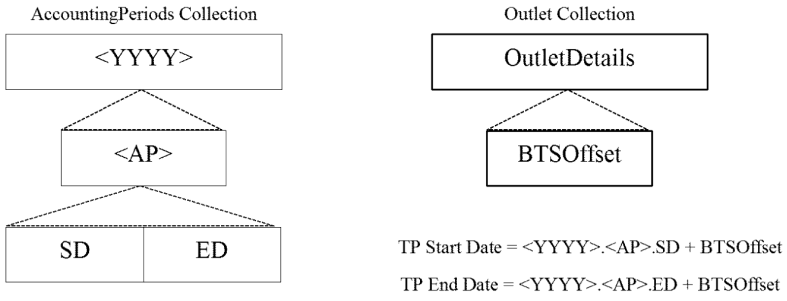


Figure 5 - TP Start/End Dates and the Accounting Period Offset

7. There is a case for preventing further TP rollovers if the current TP is already ahead of the calendar. This would avoid futile user attempts to undo a rollover by rolling yet further ahead in the vain hope of eventually getting back to where they started. However, this requirement has not been stated in [CDBT] or [DP], so will not currently be implemented.

5.1.2.6 Final CAP Rollover and Transition

5.1.2.6.1 Determining the Final CAP

To achieve the Branch-by-Branch and Stock Unit-by-Stock Unit transition to TP-based accounting, each Branch will have reference data that identifies the last CAP to be used.

Since this data is only needed to manage the transition and becomes obsolete thereafter, it is defined within a set of CAP-specific SoftLaunch definitions controlled by branch-specific products, rather than as new attributes of the branch-specific Outlet object. Where code needs to determine the value of the last CAP, it can do so by obtaining it from the SoftLaunch system attribute FINAL_CAP.

For more details, see [MIGHLD]

5.1.2.6.2 Determining the First TP

The first TP cannot simply be defined alongside FINAL_CAP as a Softlaunch system attribute because the start and end dates for a particular TP depend on the branch-specific accounting period offset. The appropriate first TP for a given FINAL_CAP must therefore be calculated for each branch.

FUJ00085124

It is desirable that this first TP is chosen such that its start and end dates include the date on which the rollover from FINAL_CAP occurs. In principle this means using the physical rollover date to determine the First TP. However, if rollover is undertaken a day or two early, the wrong TP could be chosen, causing rollover into the next TP to be required almost immediately. Another complication is that the 'NextCAP' field of EPOSSCAP.Office and StockUnits.<SU> persistent objects is filled in on **entry** to FINAL_CAP, so might be different to the actual next CAP (first TP) if that was determined on **exit** from FINAL_CAP using the rollover date.

On the other hand, determining the first TP purely from the FINAL_CAP end date could result in the wrong TP being chosen if rollover is done early or late and the FINAL_CAP end date is not aligned with an Accounting Period boundary.

Since POL have now decided to carry out the transition in four tranches, corresponding to the four branch offset values 0, 7, 14 and 21, and have chosen end date of the FINAL_CAP for each case to be aligned with the boundary between one TP and the next, the simplest option is to use the FINAL_CAP end date rather than the physical rollover date to determine the first TP.

5.1.2.6.3 Avoiding CAP/TP Ambiguity at Year-End

Since CAP and TP values are held as alternative values in the same attributes of messages and persistent objects, we need to avoid CAP/TP ambiguity when transitioning from CAP to TP mode around financial year-end.

This is particularly a problem for objects in the StockUnitMarkers persistent object collection, which are keyed on the CAP/TP number, and record rollover markers for the last 5 CAPs for each stock unit to support message store scans.

The solution is to add a bias to the TP if the normal value would clash with a value already recorded in StockUnitMarkers. The default bias is 60, and can be overridden by Type C ref data if required (unlikely).

For simplicity of implementation, this bias will be visible, so the user may see TPs in the range 61-72 (or higher) rather than 1-12 (or higher).

The effect of the bias is that CAP 1-5 will transition to TP 61-62 instead of TP1-2. For an office newly established in CAP mode, biasing may need be applied even though the first TP does not clash with an existing CAP in the StockUnitMarkers history. For example, if the office is established in CAP 3, it cannot transition to TP 1, because after two further TP rollovers, TP 3 would clash with CAP 3 which would still be in the StockUnitMarkers history.

The decision whether to bias is therefore based on what StockUnitMarkers would have contained if the office had been running for 5 rollovers, and is based on the StockUnitMarkers entry for the DEF stock unit.

Note that once a bias is established for the first TP in a given financial year, it will be used for the rest of that year to avoid further disruption to the numbering sequence, but will be removed on starting the next financial year.

5.1.2.6.4 Effecting the Transition

Stock Unit and Office rollovers will compare the current CAP against this FINAL_CAP value (if available), and if the next CAP would be beyond it, the new CAP will be set to the first TP instead of FINAL_CAP+1, and be marked as being in TP mode (see below).

FUJ00085124

Thus when a Stock Unit rolls forward from FINAL_CAP, it will enter the first TP (irrespective of the current physical date), but as described earlier, the TP value will continue to be recorded in the existing 'CAP' variables of the StockUnits and EPOSSCAP Office persistent objects (for backwards compatibility).

5.1.2.6.5 Coping with Transitional States

Prior to TP transition, the Previous/Current/Next CAP values in the StockUnit and Office persistent objects all identify CAPs, and all User Interface legends and reports must show 'CAP' legends as at present

On entering FINAL_CAP, the 'Next CAP' value will refer to the first TP, so will have the first TP value rather than FINAL_CAP+1. Any User Interface legend or report referring to that value must show a 'TP' legend, but ones referring to the Previous or Current CAPs must still show a 'CAP' legend.

On rolling forward to the first TP, legends or reports referring to the Current or Next CAP must now show a 'TP' legend, but any references to the Previous CAP must still show a 'CAP' legend.

Finally, on rolling forward into the second TP, all current legends and reports must show a 'TP' legend - but see also Historical Reports(5.1.2.7.6).

To support these decisions, the StockUnit and EPOSSCAP Office persistent objects will record additional transition status fields 'TP' and 'TPTransition'. (See Table 3 and Appendix C – Affected Persistent Object Collections). Note that these values for a Stock Unit do not necessarily match that of the Office because a particular Stock Unit will be ahead of the Office once it has rolled over to the next CAP/TP

<TP:>	<TPTransition:>	Meaning
Missing/empty	Missing/empty	StockUnit/Office is operating in CAP mode. CAP/PreviousCAP/NextCAP are all CAP-based values
Missing/empty	1	StockUnit/Office is operating in CAP mode (in the FINAL_CAP). CAP/PreviousCAP are CAP-based values; NextCAP is TP based
1	1	StockUnit/Office is operating in TP mode (in the first TP). PreviousCAP is CAP-based value; CAP/NextCAP are TP-based values
1	Missing/empty	CAP/PreviousCAP/NextCAP are all TP-based values

Table 3 - TP Transition Status Values

Note that most code is only interested in the mode for the current CAP (in order to output an appropriate legend for a prompt or report), in which case the presence of the <TP:> attribute is sufficient to determine whether operating in TP mode or not.

Code that handles rollover can cater for special cases during the transition by using <TPTransition:> to decide on detailed action. For example, the message prompt that asks whether the user wishes to roll forward into the next BP or CAP would need to be replaced by one that asks about rolling forward into BP or TP instead. However this revised question

FUJ00085124

would apply from FINAL_CAP onwards, rather than from the first TP, so the code that outputs it would need to test TP and TPTransition to determine the appropriate state.

5.1.2.6.6 Special cases

Rollover into TP from an Extended CAP should not occur, since the option to extend the CAP is to be disabled at an earlier point in the migration (see [DP] section 2.6.2). However, in the event that the time gap between Point 20 and FINAL_CAP is too short, it might be possible for an extended CAP to be set that is later than FINAL_CAP.

Resilience dictates that something sensible should be done if this happens. In particular if the designated FINAL_CAP is skipped. Similarly, we need to be resilient to conditions such as the FINAL_CAP values being missing or in the past?

The answer is to use greater than checks rather than strict equality. Thus determination of the next CAP will select the first TP if FINAL_CAP is present and less than or equal to the current CAP (including the year), otherwise it will select current CAP + 1 (subject to year-end) as now.

If FINAL_CAP is defined, but the first TP cannot be determined, transition to TP mode will be suppressed (with appropriate error message), and the branch will continue in CAP mode to current CAP + 1 (just as if FINAL_CAP had been missing).

Note that year-end does not cause a special case for transition, because week 52 is only treated specially when setting a CAP Extension. (Remembering of course comparison of current CAP against FINAL_CAP must take account of the year as well as the period).

5.1.2.6.7 Switching between TP and CAP Modes

During the period where one Stock Unit has rolled over into TP mode, but another is still operating in CAP mode, it may be necessary for the user to switch between TP-based menus and CAP-based menus – e.g. whenever they switch Stock Units (by logoff/logon or by explicit Stock Unit reattachment). This is handled by invoking SoftLaunch dynamically to update the desktop menus according to the state being entered. See section 9.1.2 and [MIGHLD] for more details.

FUJ00085124

5.1.2.7 User Interface and Report Changes

Changes must be made to the User Interface and reports in order to replace 'CAP' by 'TP'. . The following situations need to be dealt with:

- 'CAP' (or similar) text is burnt into code
- 'CAP' (or similar) text is burnt into Global Objects
- 'CAP' (or similar) text is burnt into Reference Data
- 'CAP' (or similar) text is burnt into persistent objects and output more or less transparently

The following sections consider the changes, and identify techniques to handle these cases, bearing in mind that

- S80 code and reference data must accurately emulate pre-S80 systems until new behaviour is explicitly activated as part of the migration strategy described in Migration (9.1) and [MIGHLD].
- During the transition the current Stock Unit may be in a different mode to that of the Office or other Stock Units, which affects the decision on whether 'CAP' or 'TP' should be output.

5.1.2.7.1 Desktop Status Area

When the desktop is idle for a logged-in user, the status area on the right-hand side identifies the current CAP for the Stock Unit to which the user is attached.

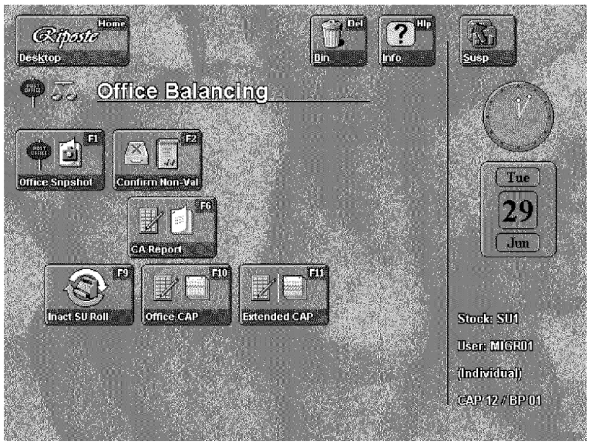


Figure 6 - Example of CAP legend on menu button and status area

The 'CAP' legend in the status area is currently output by hard-coded logic in the EPOSSStockUnit code whenever the Stock Unit is updated. Since the decision to output CAP or TP depends on whether the particular StockUnit has yet rolled into TP, and that fact is recorded as a property of the StockUnit persistent object, the code simply needs to test that property before outputting 'CAP' or 'TP' as appropriate. The actual 'CAP' or 'TP' text will now be taken from the appropriate attribute of the Reference Data 12.3 (see Appendix B – Affected Reference Data Collections)

FUJ00085124

5.1.2.7.2 Reports

Reports are composed from common report sections included in sequence, which in turn contain item definitions. Some item definitions have the string CAP (or CAP-related text) burnt into them.

The section definitions are provided as global object data in the EPOSSRSect collection (see Appendix B – Affected Reference Data Collections). Access to global objects is achieved similarly to reference data in Riposte message store, (and indeed at one time global objects were just that). However, they now have separate implementation on top of the datafile 'globalobjects.dat', and this file is shipped at the same time as code. Therefore, it is OK to replace hard-coded CAP legends by infills because updated global objects do not have to work with pre-S80 code.

For example, the report header object 4601 ('Office Header 40 cols') contains the item definition 'CAP Label', defined as plain text, with a parameter value of 'CAP'., and the item 'CAP', defined as a system variable infill, with a parameter value of 'OfficeCAP'

<I:<N:CAP Label><IT:PlainText>...<O1V:CAP:>...>

<I:<N:CAP><IT:SystemVariable>...<O1V:OfficeCAP:>...>

The PlainText legend item can be changed to be of type 'SystemVariable', with a single parameter of (say) 'OfficeCAPLegend', but because of the transition period when the current Stock Unit is operating in TP mode but the Office as a whole is still operating in CAP mode, the choice of CAP/TP legend depends on whether the report is for the current TP-mode StockUnit or not.

That is not in itself a problem, because distinct report sections are already used for those cases, and the clue is in the system variable used to provide the actual CAP/TP value – OfficeCAP retrieves the CAP for the Office, whereas CurrentCAP retrieves the CAP for the StockUnit. However, in some cases the choice of CAP/TP is based on parameters passed rather than context information.

The general solution approach is to introduce a distinct legend type for each value type. Thus the SystemVariable OfficeCAP would be paired with a new SystemVariable OfficeCAPLegend, and the PassedParameter SelectedCAP would be paired with a new PassedParameter SelectedCAPLegend. The internal details of how these fields are to be supported by the Report Broker and its clients will be covered in the low level design, but the context information to support any TP/CAP decision will be based on the TP transition status for the StockUnit or Office as appropriate (see Final CAP Rollover and Transition (5.1.2.6)). See also Historical Reports (5.1.2.7.6) for the special case of 'SelectedCAP'.

Another issue is the reporting of Cash Account week numbers. In the steady state TP system, those reports are obsolete, so there is little point in doing a perfect job of mapping them onto TP numbers. For example the Cash Account report contains a heading 'CA Header 7' contains plain text 'week ending'. This could (in principal) be converted to say something like 'period ending' in the same way as CAP is replaced by TP, but it is not worthwhile because the entire report will become obsolete, so it will not be changed.

5.1.2.7.3 Desktop Menu Buttons

Some desktop menu buttons have CAP-related text burnt into them. Others need to be revealed or suppressed according to whether the Stock Unit has been rolled into TP yet or not (or indeed whether S80 functionality is yet to be made visible).

FUJ00085124

The button definitions for particular menus are provided as objects in temporal Reference Data collections called 'DesktopButtonsIItem<n>' in Riposte message store, and the required effect will be achieved by use of [SOFTLAUNCH].

On attaching to a Stock Unit (e.g. during logon), EPOSS will invoke SoftLaunch dynamically in order to refresh the desktop buttons with ones suitable to the CAP/TP mode of that Stock Unit. On rolling a Stock Unit or Office into TP mode, EPOSS will similarly invoke SoftLaunch dynamically in order to refresh the desktop buttons for the new mode.

See section 9.1.2 and [MIGHLD] for more details.

5.1.2.7.4 Messages

Some User Interface prompts and messages have CAP-related text burnt into them. These messages are provided as objects in the 'MessageDefs' temporal Reference Data collection in Riposte message store.

For example, message 58 (MSG_ROLLINTONEXTCAP) contains the text

"Do you wish to roll over into the next Cash Account Period (%NextCAP%) or into the next Balance Period in this CAP (%ThisCAP%/%NextBP%)?".

Figure 7- Example of Message containing CAP-specific text

Rather than replace 'Cash Account Period' by a new infill %CashAccountPeriodLegend% (say), which would cause problems if used with pre-S80 code, the solution is to provide alternative messages, and make the S80 code select the variant according to whether the current Stock Unit or Office is operating in TP mode, as applicable (see 13.1). Note that this does not involve SoftLaunch, because the messages are output by code directly.

See the MessageDefs Collection (Appendix C – Affected Persistent Object Collections) for a list of CAP-related messages that need to be handled in this way.

5.1.2.7.5 Report Criteria

Some report criteria have CAP-related text burnt into them. The criteria definitions are provided as objects in the 'RepCriteria' temporal Reference Data collection in Riposte message store.

FUJ00085124

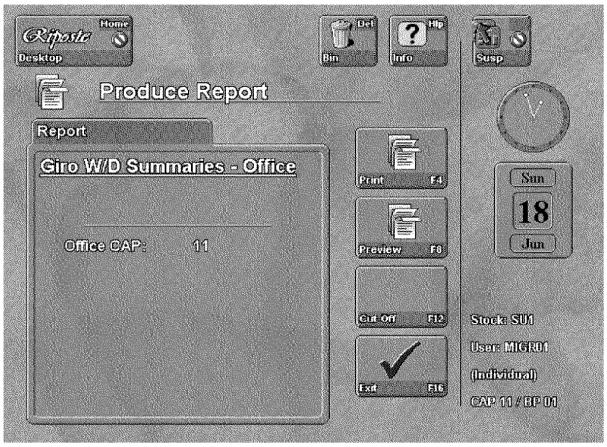


Figure 8 – Example of CAP Legend in Report Criteria

For example, the report tablet in Figure 8 shows an 'Office CAP' legend by virtue of the report referring to report criteria object 9, which has value

<RData:<Data:<Name:OfficeCAP><Prompt:Office CAP><Type:OfficeCAP>>>

Since such criteria are shipped as reference data, they have the same problem as messages - hardcoded 'CAP' texts cannot be simply replaced by 'TP' or new infills, because they would not work with pre-S80 code.

A blanket replacement of 'CAP' by 'TP' would give the wrong effect during transition – a report for a Stock Unit that has rolled into the first TP should have a TP legend, but one for a Unit that is still in the final CAP should retain the CAP legend. The solution is to select the appropriate criteria object alternative according to its 'Type' attribute and the StockUnit/Office transition status. Thus when used with pre-S80 code, or post-S80 code for a Branch or Stock Unit still operating in CAP mode, the original CAP-only criteria would be selected.

For the criteria selections that are specified as part of the desktop buttons for the appropriate Reports submenu, this will be achieved by use of SoftLaunch, as described for Desktop Menu Buttons (5.1.2.7.3).

For example, if the Stock Unit is operating in TP mode, but the Office is still in CAP mode, then desktop buttons relating to reports for the current Stock Unit should be made to refer to TP criteria variants, but those relating to reports for the Office (as in the example above) should be made to refer to the pre-S80 CAP criteria variants.

However, some reports specify the criteria selection dynamically (see Historical Reports (5.1.2.7.6). For these, the code logic must check the mode of the relevant period and select the CAP/TP criteria accordingly.

5.1.2.7.6 Historical Reports

When requesting a report reprint or other report that is not specific to the current StockUnit or Office context, the user must specify the particular period. In some cases this is by physical date range, but in others it is by a particular CAP.

FUJ00085124

This is currently done by presenting a picklist of available CAPs, based on entries in the 'StockUnitMarkers' collection (if related to a particular StockUnit) or the 'OfficeRePrints' collection (if related to the entire Office), but the picklist entries are generated to be of the form 'CAPnn' or 'CAPnn-BPn'. This would be a problem if the output text should instead be 'TPnn' or 'TPnn-BPn' once transition to TP mode is complete. Even worse, during the transitional period, there would need to be a mixture of CAP and TP texts presented.

However, [DP] section 2.5.1.3.8 indicates that these reports will be changed to report a date range, so will no longer need a CAP-based picklist entry during period selection.

The current approach to date selection is by direct user entry of physical start and end dates. While this may remain appropriate for ad hoc event log reports, it makes little sense for reprints, since only particular dates are available. Therefore, a picklist will be required for these, showing the date range for each report to be reprinted rather than the CAP.

For reports that cover a physical date range, the individual entries are often described in terms of the CAP/BP in which they occur. For example, the event log report in Figure 9 has 'CAP', 'BP' and 'NODE' column headings applicable to event descriptions for the periods CAP 8 and CAP 11.

```
RDDT                              FAD: 9017777
11:38 12/07/2004     CAP:15  BP:02  SU:SU1
Event Log: Balancing - Office Copy




SU  USER           CAP BP NODE
DATE AND TIME    EVENT TITLE
EVENT DESCRIPTION

SU1 MIGR01          08 02 01
25/06/2004 10:03 Declaration Complete
ONCH Total £0.00

SU1 MIGR01          08 02 01
25/06/2004 10:03 Declaration Complete
DeclareCash Total £0.00

SU1 MIGR01          08 02 01
25/06/2004 10:04 Declaration Complete
DeclareStamp Total £0.00

SU1 MIGR01          11 01 01
25/06/2004 10:21 Rollover Complete
SU SU1 rolled : CAP 8 BP 2 to CAP 11 BP 1
```

Figure 9 - Event Log Report Spanning Multiple CAPs

If the range of periods being reported span the transition from CAP to TP, then the column headings will either need to be changed as the boundary is crossed, or be inaccurate before or after the transition.

Since the most useful information is actually in the report row (e.g. 'SU SU1 rolled : CAP 8 BP 2 to CAP 11 BP 1') rather than the column headings, it is proposed that the heading will

FUJ00085124

simply be changed to 'TP' if the Office or StockUnit has now entered TP mode – i.e. the column headings will be in step with the overall report heading rather than the content of particular rows.

5.1.2.8 MiMAN Migration

There continues to be a requirement to introduce new offices to Horizon.

Introducing Horizon to branches requires migration strategies from any prior systems. The two existing strategies are MiECCO (for branches previously operating the ECCO system) and MiMAN (for branches operating a manual system, or starting from scratch).

There is no requirement for MiECCO to support migration to Horizon in Branch Trading mode, because the ECCO-based branches have already been migrated to Horizon (operating in Cash Account mode). The MiEcco service has already since been withdrawn.

Whilst originally conceived to migrate existing branches operating manual systems to Horizon, MiMan has continued to be used as the vehicle to support the opening of new branches. Such a circumstance needs to occur, for example, following temporary closure or in the case of a temporary office such as that opened for Wimbledon fortnight.

As part of Impact Release 3 a new version of MiMan will be introduced. It will support the creation of new branches from migration point 20 and hence will support both new branches introduced and operating using Cash Account Periods, and new branches operating trading periods.

The variant of MiMan will operate differently from the existing application in the following key areas:

- The current main purpose of MiMAN is to capture the initial assets, using Cash Account mappings to formulate dynamic data capture screen dialogues. As part of Impact Release 3 the migration of a new branch will merely provide a checkpoint of migration completion. Any initial assets of the office must then be introduced using remittance transactions.
- The migration report produced as part of MiMan becomes obsolete as there is no captured data to report.
- The migration of a new branch needs to address whether the office is to be migrated into a Cash Account Period or a Branch Trading Period. Branches will all be migrated to Cash Account Periods prior to point 30, and may be migrated to either beyond that to point 50.

A new application is required to be developed as part of Impact Release 3 to replace the existing MiMan application at Point 20 in the migration path. It will be initiated by desktop button in the same way as the existing manual migration. Activation of the new application will be driven by softlaunch controlled by existence of the S80 codebase.

The new application will be based on the existing MiMan program, however all the data capture functionality will be removed as will the migration report.

The application will use mechanisms defined in 5.1.2.6 to determine whether the branch is to be migrated into either a CAP or TP. The user will no longer have the option to override the pre-determined value.

FUJ00085124

All other processing of the MiMan variant will remain as undertaken by Miman; the branch will be migrated into BP2 of the chosen accounting period.

It will be necessary however for the application to prepare further opening figures if the branch is migrated to a TP. The final Stock Unit and Office Rollover figures prepared when moving from CAP to TP prepare additional information for BTS production. Whilst no actual data is required any mandatory preparation of figures must be undertaken.

It is proposed that the current Administration Menu Button F3 Migration simply becomes an application to initiate and complete migration. It will engage a much simpler dialogue than the migration process demands at present.

Note that the ability to create stock units before migration is complete must also be inhibited.

Thus the new migration application launched from the Administration menu will:

- Determine the appropriate CAP or TP from calendar and Impact Transition information.
- Checkpoint the branch to designate the branch as migration complete.
- Put up a dialogue advising migration complete, and state the accounting period into which the office is migrated.
- Re-enable creation of stock units

FUJ00085124

5.1.3 Introduction of Aggregation Engine for Volume Stock

The handling of Stock By Volume for Impact Release 3 requires changes to the Balancing Figures Aggregation. The main visible impact of this is on the Stock Unit and Office Balancing Reports the format of which will change to accommodate stock by volume rather than by value. The introduction of monthly rather than one week accounting periods also has an impact on the aggregation of figures as a larger volume of data will contribute to the balancing process performance.

Aggregation of figures for most balancing and reporting tasks is the responsibility of a component of EPOSS, EPOSS Data Server.

5.1.3.1 Data Server Architecture

Simplistically Data Server provides an aggregation engine that acts in response to client application invocations to summarise raw transaction data into a hierarchic aggregation structure specified as reference data and known as the EPOSS Accounting Node Hierarchy. Data Server then exposes APIs which may be utilised by the client application methods to retrieve the summarised data as required for reporting or the preparation of rollover figures. The Node Hierarchy represents the way in which summarised transaction data is presented for analysis by the Post Office on reports. Hence the aggregation structure is owned by the Post Office. The architecture is depicted in the following diagram.

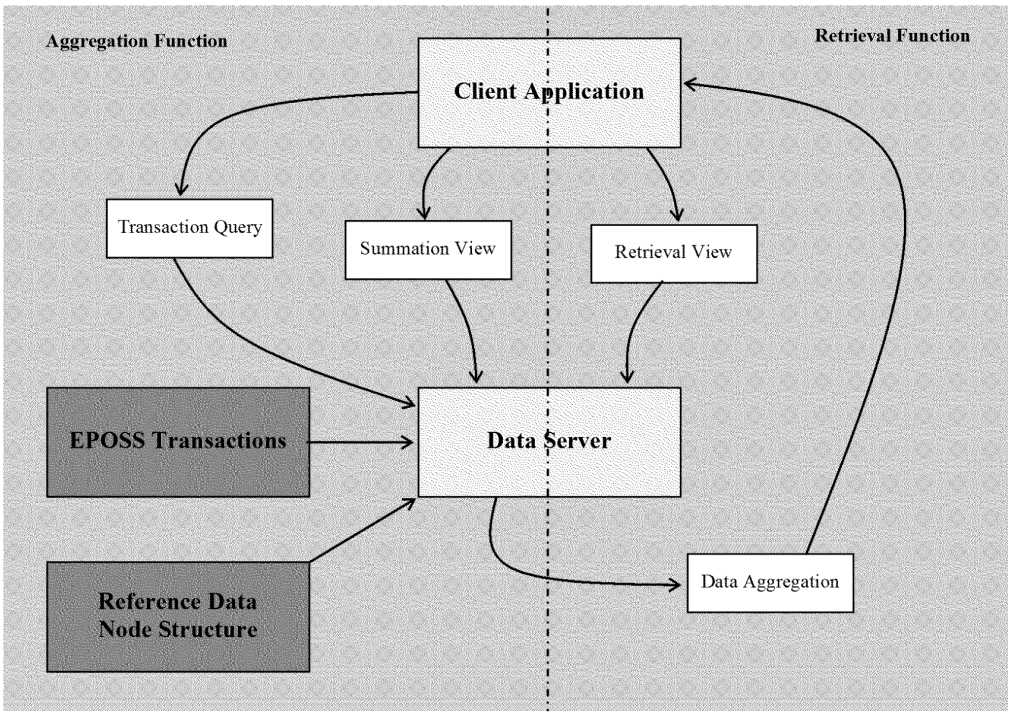


Figure 10 – Data Server Architecture

FUJ00085124

Data Server can be seen as providing two logical functions to client applications:

- Aggregation Function
- Retrieval Function

The Aggregation Function supports the Client Application passing parameters in the form of a Query that defines the transactions to be taken into account in compiling that aggregation, and a Summation View that defines what attributes are to be aggregated and how. Data Server applies accumulation rules, for example summing attribute values or counting attribute occurrences, to each transaction satisfying the Transaction Query to produce data aggregation figures for each item in the summation view. The resulting data aggregation is available in memory for the client application to use, whether to process summary figures or for reporting.

The Retrieval Function exposes APIs to the Client application in order that it may invoke retrieval methods that access the accumulated data, returning values for reporting or further processing.

The following overview of the primary elements of data server describes the basic architecture of data server.

5.1.3.1.1 EPOSS Transactions

The raw data aggregated by Data Server is furnished from messages previously posted to the Riposte Message Store, filtered by conditions specified in the Transaction Query. Data Server is primarily concerned with the aggregation of figures for branch reporting and balancing. Hence the main category of message forming the raw data are EPOSS Transactions.

5.1.3.1.2 Node Structure

The Data Aggregation structure is provided by Reference Data. The Collection <EPOSSNodes:> first provides the objects which define a static structure. The static Node Structure provides the post office view of how ALL transaction postings are to be analysed in branches across the entire network. There are two such structures modelled in EPOSSNodes at present. The first is the Accounting Node Hierarchy and the second is the Cash Account Node Hierarchy. The Cash Account Node hierarchy will cease to be maintained after all branches have migrated to S80.

The Accounting Node Hierarchy is the vehicle which defines, for example, that movements of cash shall be analysed as a total, within the category (node) of Methods of Payment, and that Methods of Payment form part of the overall stock total, within the total office balance.

At each level of analysis, defined by a node, a number of accumulators are defined, Each accumulator, or bucket, defines what items of accumulation are supported by that specific node. Currently the following accumulators are defined:

- Sale Value (sv)
- Quantity (qty)
- Number of Transactions (rc)

A more expansive explanation of the Accounting Node Hierarchy is provided in 15.0. The static Node reference data specification is provided in Appendix B – Affected Reference Data Collections.

FUJ00085124

The Accounting Node hierarchy will be extended as a result of changes to the way by which transaction data is aggregated as a result of S80.

In order to support products defined as stock to be managed by volume the set of nodes forming the Accounting Node Hierarchy will be extended. New sub-branches of the hierarchy will be defined under the Receipts Node definitions. These new sub-branches to the structure will provide the nodes against which sale and adjustment transactions of volume stock will be recorded, and support the concept of Tertiary Mapping explained later. As volume stock transactions no longer contribute to value stock movement nodes of the structure, yet their sale and adjustment transactions need reporting on balance reports these additions to the structure are necessary.

Additionally new Accumulators will be defined against the existing Value Stock Node sub-branch structure. These new Accumulators will allow conditional accumulation of transaction attributes, their purpose being to accumulate based on whether the contributing product is volume stock or not. The provision of these new accumulators will allow the total value stock nodes to represent a figure which excludes volume stock, even though transactions still contribute to the hierarchy by virtue of their primary mapping.

5.1.3.1.3 Transaction Query

The transactions to be taken into account in any one accumulation are provided to Data Server by the client application as a Transaction Query in the form of a criteria string. The string will contain a set of attribute name, operator and value conditions that together can match EPOSS Transaction attributes of the transactions to be considered. An example string could be:

sCriteria = "<SelectExpression:(EPOSSTransaction.SM.L5 EQ ""3017""
OR EPOSS Transaction.PM.L5 EQ ""3017"") "
& "AND ((NOT Exists(EPSSTransaction.OpeningFiguresId))
OR (EPOSS Transaction.OpeningFiguresId EQ """"))) "

Also provided will be a pair of bound values which specify the lower and upper markers of transactions to be considered, so as to avoid unnecessary processing of the entire message store. Ordinarily the lower marker is likely to be the last rollover marker and the upper marker will be the current latest message. Upper Markers are more often provided for specific use in undertaking report reprints.

5.1.3.1.4 Summation View

The set of transactions to be aggregated in any one summarisation is defined by the transaction query. The static structure into which the data is aggregated is provided by the Reference Data Collection <EPOSSNodes:>. Any transaction posted shall be capable of being aggregated into this static structure.

There is also to be considered though, the concept of a dynamic structure. For example a particular report may examine transactions as a group, by Stock Unit within Balance Period. In other words whilst the analysis structure is static the summarisation of transactions within that structure is dynamic depending on the needs of the client, and what transactions are encountered.

FUJ00085124

The summation view provides a specification (or template) of how the transactions themselves are to be grouped and what attributes within or about the transaction are to be used to accommodate that grouping. Again further explanation is provided in 15.0.

The summation View is provided by specifying an object reference to another collection of reference data, <EPOSSDNodes:>.

A DNode, or Dynamic Node, specifies a dynamic node structure, providing the template to be used to group (and order) transactions for a particular report (client use). As data server builds up the data aggregation it compiles a dynamic picture of nodes (now dynamic nodes) representing the transactions as an instance of the picture provided by the dynamic node template. As each transaction contributes to this dynamic picture if its position within the dynamic structure requires a new instance of one of the groupings then dynamically data server creates a new grouping instance. So for example as a minimum data server will have a dynamic node instance for each product, aggregating all transactions transacting that product.

For each transaction selected in the Transaction Query, like the static node structure, the DNode specifies what actions are to be taken to aggregate that transaction's data into the dynamic picture. A typical Dynamic Node is specified in 12.0.

Again, the following attributes may be aggregated:

- Sale Value (sv)
- Quantity (qty)
- Number of Transactions (rc)

5.1.3.1.5 Data Aggregation

The outcome of the Aggregation Function is a Data Aggregation in memory, defined in a structure the template of which is provided by EPOSSNodes, together with EPOSSDNodes. The content of the aggregation is compiled from the application of accumulation rules which satisfy the summation view, to transaction attributes of those transactions which meet the criteria in the given Transaction Query. A representation of the aggregation picture is provided in the following diagram:

FUJ00085124

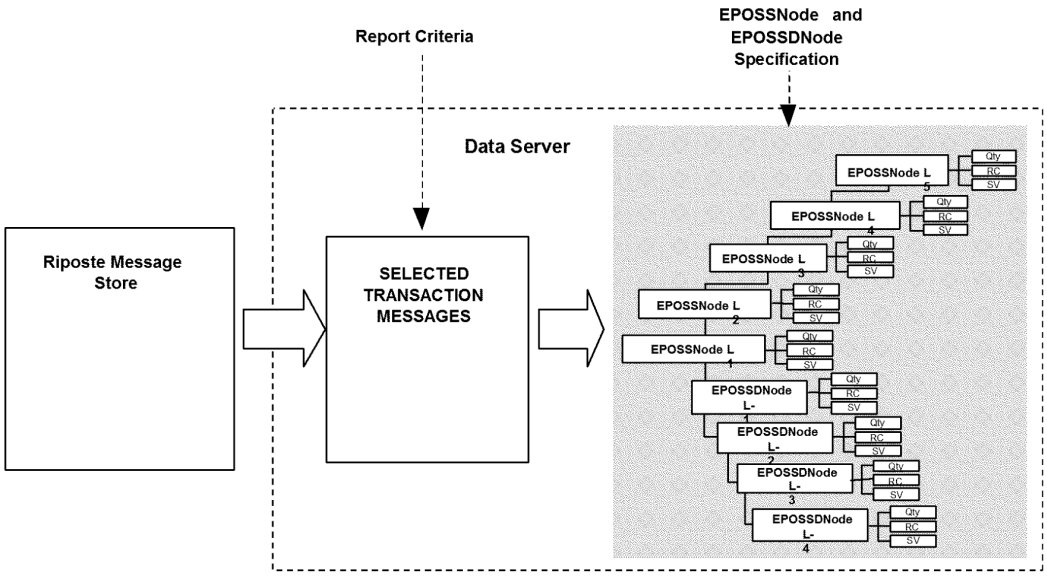


Figure 11 – Data Aggregation

5.1.3.1.6 Retrieval View

The aggregated data is made available back to the client application by data server exposing API's to the client. A defined set of retrieval methods may be invoked to return the details of the aggregation.

5.1.3.2 Accumulation Rules and Accumulators

Crucial to the changes required of Data Server is an understanding of the inner processes involving the concepts of accumulation rules and accumulators. From the outset it was stated that data server applies accumulation rules to populate the accumulators defined in the node structures, whether static or dynamic.

5.1.3.2.1 Accumulators

Both the static and dynamic accounting node structures support the concept of an Accumulator. An accumulator is an aggregation 'bucket' and is associated with a specific Node (noting that this can be a node from the static hierarchy or built dynamically). The node defines a specific level of business analysis eg. the aggregation of data about MOPs is contained in Node 3003, and each accumulator provides the specific items of information that can be gleaned from the aggregation of data in that node.

Currently three accumulators are defined:

- Sale Value (sv), the total of all transaction sales analysed by node
- Sale Quantity (qty), the total volume of transaction product sale quantities analysed by node

FUJ00085124

- Number of Transactions (rc), the total number of transactions contributing to the node

5.1.3.2.2 Accumulation Rules

The Accounting Node Hierarchy provides a structure into which transaction information is aggregated or summarised. As stated each Node provides a bucket, or 'pigeon hole' into which corresponding transactions can be 'mapped'.

The relationship between the static node hierarchy and transactions is made by the attributes of the Primary Mapping (PM:) and the Secondary Mapping (SM:), if it is present, in each transaction. A new Tertiary Mapping (<TM:>) is now conditionally added to each transaction.

The primary mapping is a mandatory attribute of each transaction and provides the transactions' position within the hierarchy for the purposes of accumulating debit and credit value of all transactions entered.

The secondary mapping is a conditional attribute of transactions. It is assigned to a transaction based on the specific mode the transaction is conducted in. Not all modes have secondary mappings. Its use relates entirely to the movement of stock and it provides, in total, the total movement of stock in a particular mode.

The tertiary mapping is a conditional attribute of the transaction. It is assigned to a transaction based upon whether the product being transacted is a volume stock product and on the mode the transaction is conducted in being a specific mode.

Transactions which are accumulated will comprise both product movements and Opening Figures, specified by the transaction query criteria. Each transaction that satisfies the selection criteria of transaction query will have its attributes accumulated into the aggregation based on the presence of the primary, secondary and tertiary mappings, as follows:

For the Primary Mapping,

> *Accumulate values of the sale value, quantity and Number of Transactions into each node identified by the nodes in the PM:, according to the presence of the appropriate accumulators.*

For the Secondary Mapping

> *If the transaction has a Secondary Mapping, accumulate values of the sale value, quantity and Number of Transactions into each node identified by the nodes in the SM:, according to the presence of the appropriate accumulators.*

For the Tertiary Mapping

> *If the transaction has a Tertiary Mapping, accumulate values of the sale value, quantity and Number of Transactions into each node identified by the nodes in the TM:, according to the presence of the appropriate accumulators.*

The accumulation rules are independent of the accumulators; in other words the accumulation rules are applied to all accumulators. Currently the accounting node hierarchy defines all nodes to contain all accumulators, hence given any one aggregation, all three attributes are accumulated.

5.1.3.3 Changes to Data Server

The Data Server architecture will remain largely unchanged. Two primary changes are required to the Data Server to support handling stock by volume. The existing version of Data

FUJ00085124

Server will be enhanced. Initially it was considered that a copy of data server should be taken and a new version developed so that any risk of error could be isolated. Analysis of the changes required to data server however has revealed that the changes are minimal and not invasive. Thereby the risks of error associated with these changes are limited, and when weighed up against having to maintain two versions of the component, justification for enhancing the existing version of data server is made. The enhanced Data Server will be adopted for all balance reports, including the Office Snapshot.

5.1.3.3.1 Introducing Conditional Accumulators

The first change to data server is required to support stock handling by Volume. This necessitates the transaction rules will change.

A new concept is introduced - the conditional accumulator. A conditional accumulator combines the introduction of a new accumulator, with an accumulation rule, defined by a new Accumulator Function. In other words unlike existing accumulators which have the same accumulation rule applied to all nodes to which the transaction maps, a conditional accumulator will only have data aggregated if the rule is satisfied. Their purpose is to accumulate an attribute from contributing transactions based on specific conditions being met, so as to preclude, or exclusively include, transactions that make up value stock, excluding products that have been re-nominated as volume stock.

The following conditional accumulators are introduced. The accumulators are specified against the appropriate Reference Data Node Item. Not all accumulators will apply to all nodes.

Accumulator	Function	Condition Rules
SVETM	SumETM	If Transaction Product has a Tertiary Mapping (TM:) attribute value, then accumulate Sale Value into accumulators for nodes identified by transaction mappings. In other words when applying the Primary and Secondary Mappings of a transaction, if the transaction product is denoted as volume stock this accumulator will be applied. This accumulator is redundant if applied to a Tertiary Mapping Node
QtyETM	SumETM	If Transaction Product has a Tertiary Mapping (TM:) attribute value then accumulate Quantity into accumulators for nodes identified by transaction mappings. In other words when applying the Primary and Secondary Mappings of a transaction, if the transaction is denoted as volume stock this accumulator will be applied. This accumulator is redundant if applied to a Tertiary Mapping Node
RCETM	CountETM	If Transaction Product has a Tertiary Mapping (TM:) attribute value then increment accumulators for nodes identified by transaction mappings. In other words when applying the Primary and Secondary Mappings of a transaction, if the transaction is denoted as volume stock this accumulator will be incremented by 1. This accumulator is redundant if applied to a Tertiary Mapping Node. A non-zero value of the accumulator indicates that volume

FUJ00085124

		stock products mapping to this node have been transacted, and can be used to decide whether the node should be reported in the 'Volume Stock' section of a Stock Unit Balance Report.
SVNTM	SumNTM	If Transaction Product has no Tertiary Mapping (TM:) attribute value then accumulate Sale Value into accumulators for nodes identified by transaction mappings
QtyNTM	SumNTM	If Transaction product has no Tertiary Mapping (TM:) attribute value then accumulate Quantity into accumulators for nodes identified by transaction mappings
RCNTM	CountNTM	If Transaction product has no Tertiary Mapping (TM:) attribute value then increment accumulators for nodes identified by transaction mappings. A non-zero value of the accumulator indicates that value stock products mapping to this node have been transacted, and can be used to decide whether the node should be reported in the 'Value Stock and MOP' section of a Stock Unit Balance Report.

Table 4 - Conditional Accumulator Types

A Tertiary Mapping is a new attribute introduced onto transactions which transact stock, that is now to be handled by volume rather than value. The existing Accumulators will continue to be applied without change. The Tertiary Mapping Attribute, <TM:>, is in exactly the same format as PM: and SM:.

The changes required satisfy the proposals made in [DP] at 2.5.1.4.

5.1.3.3.2 Improving Data Server Performance

Secondly, to support balancing by trading period the volume of transaction data to be processed in any one accounting period is likely to increase substantially. Whilst the promoted use of Balance Periods will assist in minimising loss of accounting control over the longer duration of Trading Periods the amount of data to be processed means that performance may become an issue.

The addition of the new accumulation functions and the presence of tertiary mappings on many transactions compounds this problem.

To this end, the opportunity to improve Data Server Performance will be taken in two specific areas. There is no stated requirement to improve performance of counter functionality. Equally however performance must not be degraded. The increased workload to maintain information about stock by volume necessitates data server performs additional work.

A Forecast Change has also been specified (see 0.6 Changes Expected) in the event of continued problems with Data Server performance, and in particular the ability to produce balance reports without any degradation in performance as a result on the changes required for Impact Release 3. The forecast change recognises that performance can be substantially improved with a change to the architectural approach given to transaction accumulation, though the change does come with some risk.

FUJ00085124

Any invocation of data server to aggregate a set of transactions can be divided into three phases:

- Initialisation, a one off task that includes the building of the static accounting node hierarchy template in memory from reference data. In other words an empty tree in memory is built
- Retrieval Cycle, the main part of the aggregation, to process all transactions satisfying the selection criteria.
- Termination, a set of tasks to end the aggregation

Attention to the following non-invasive changes will assist in ensuring that data server performance is not degraded:

- Remove the static accounting node hierarchy tree build as part of initialisation unless balancing is being performed
- Remove unused or misused accounting node hierarchy node accumulators in reference data

Removal of static tree build

An empty aggregation tree is built by data server as part of initialisation. This means that a complete tree is built regardless of what nodes will actually be needed by the transactions needing aggregation.

The static tree build as part of initialisation will be removed and replaced by a dynamic tree build as part of retrieval cycle, building only those elements of the tree that are required by transactions being aggregated.

Removal of reference data accumulators

The static accounting node hierarchy defined by the reference data collection EPOSSNodes essentially defines three accumulators for each and every node currently. They are sv, qty and rc as previously introduced.

Examination of what those node accumulators would represent, should transactions be summarised into them, reveals their semantic to be flawed or meaningless. The use of such accumulators to report transaction summary information would provide results of little meaning.

By removing those accumulators that have no use the reference data definitions can be simplified and performance of data server improved. The following changes to node accumulators will be implemented in reference data. It should be noted that the new accumulators will only be defined on those nodes where they are required from the outset.

- All accumulators will be removed from the static root node 3017
- All qty accumulators will be removed from all Level 5, 4 and 3 Nodes

FUJ00085124

5.1.4 Amendment of Transaction Attributes for Volume Stock

The handling of Stock By Volume for Impact Release 3 requires changes to the way transactions are recorded, and in the case of stock movement (remittances and transfers), changes to the value of the transaction movement. The main visible impact of this is on the transaction stack when products are transacted as a movement.

Four changes are required to the way transactions are conducted:

- Products that are to be handled by volume henceforth will have a new attribute associated with them, a Tertiary Mapping, <TM:>. In order to be able to record the sale of such products correctly, the <TM:> attribute will be assigned a value in transactions recorded for such products.
- The movement of volume stock products, transacted in remittance or transfer modes, requires that the transaction value will be recorded as zero.
- The settlement of all transactions will henceforth be handled by POL Products rather than POA products.
- Transactions recorded by APS must be posted via EPOSS Core in order to implement all changes to transactions structures resulting from the introduction of volume stock

Consequential to the introduction of handling stock by volume rather than by value is the proposal to merge the resulting volume stock with non-value stock.

5.1.4.1 Identification of Volume Stock

The information required to identify those products that are to be managed as stock by volume as opposed to value, is all provided by POL Reference Data.

The EPOSSProducts Collection ([RDMCHLD]) currently defines all Value stock by the attribute <I:True>. Henceforth the <I:> attribute shall be deemed to define "Inventory" Stock if set to True. In other words all stock that is controlled as part of the branch inventory is identified by <I:True>

Inventory Stock can be defined as either Value Stock or Volume Stock as a result of Impact Release 3. As a result a new attribute as part of each EPOSSProducts Object is defined. The attribute is <VolS:> and takes a value of True or False. A value of True indicates the stock is volume stock, and a value of False indicates the stock is Value Stock.

All Product Objects defined as Inventory Stock and as Volume Stock will have an additional attribute <TM:>, a tertiary mapping. The tertiary mapping is defined in exactly the same format as a primary mapping (<PM:>) or secondary mapping (<SM:>) and represents the mapping of the product into the receipts or payments table, when transacted other than as a movement, normally serve customer. The primary mapping attribute is still defined for Volume Stock Products.

New elements to support Tertiary Mapping nodes will be set up as EPOSSNodes Collections as described in 12.4.

The following table summarises.

<I:> Value	<VolS:> Value	<TM:> Defined	Product Set Defined
False	Not Relevant	No	Non-stock Products, eg. Receipts or Payments Products

FUJ00085124

True	True	Yes	The set of Stock Products managed by Volume
True	False	No	The set of Stock Products managed by Value

5.1.4.2 Changes to all Transactions

EPOSSCore will be amended to record the new attributes <TM:> and <VolS:> as part of all transactions. This is because they are used by DataServer when accumulating transactions in bulk into appropriate accumulators, so need to be easily available without incurring the overhead of retrieval from the product definition each time.

The attributes will however only be populated under certain conditions as follows.

The changes required are made to EPOSSCore and are driven by new reference data.

The <TM:> attribute will be populated if the Product being transacted represents volume stock and is made as a sale or adjustment (or not a transfer or remittance). In such circumstances when the product is sold or adjusted the sale value contributes to the receipts and payments totals for balancing rather than the value stock total, as was prior to Impact Release 3. As a result the Tertiary mapping replaces the Primary mapping as the vehicle to analyse the value of sales of such volume stock products.

The <VolS:> attribute will be populated if the Product being transacted represents volume stock regardless of the mode the product is being transacted in, and whether or not the Stock Unit is operating in TP mode (PEAK 117079).

The processing changes to EPOSSCore are defined as follows.

When transacting a product the <VolS:> attribute on the transaction is always taken from the <VolS:> attribute on the EPOSSProducts Collection Object ([RDMCHLD]) corresponding to the product being transacted.

Further when transacting a product if the product is defined as volume stock, and the attribute <VolSValue:> on the <Collection:ModeParameters> object in 12.0 corresponding to the mode being undertaken does not have a value <VolSValue:Zero>, then the <TM:> attribute from the product reference data object is used to populate the <TM:> attribute on the transaction. The processing required is complicated when the Mode in question is ER (Existing Reversal).

In a mode of ER the attribute <VolSValue:> will be examined not on the object within <Collection:ModeParameters> for ER, but on the object within the collection of the original Mode of the transaction being reversed (PEAK 117784).

The need to check attributes on the ModeParameters Collection is required to ensure the tertiary mapping is only applied to transactions where the product is sold or adjusted and not moved, a transfer or remittance. Another way of looking at this is that Tertiary mappings provide the position of the product in the receipts totals for analysis purposes. If Tertiary mappings are applied in other modes than sale and adjustment then the receipts and payments quantities will not reflect the quantities of such products sold as they would contain movement transaction totals too. In the case of an existing reversal it is equally important to exclude any reversal of a transaction that is excluded now being reversed, but include reversals of transactions where the original transaction would be included.

FUJ00085124

There are migration implications of this change. The introduction of the attribute <VolS:> and the determination of its value will be implemented at migration point 20. However the introduction of the attribute <TM:> and the assigning of tertiary mappings to it will only be implemented when a stock unit has migrated to accounting under Trading Periods rather than CAPs, in other words at migration point 50.

5.1.4.3 Changes to Stock Movement Transactions

When transactions conducting a stock movement, remittance or transfer, are undertaken where the product is being handled by volume rather than value then the products will be transacted in such modes with a zero sale value. As a consequence the stock is 'moved' by quantity rather than value. Similarly when such products are transacted as an adjustment then the products will be transacted in such modes with a loss sale value.

The changes required are made to EPOSSCore and are driven by new reference data.

When transacting a product, if the product is defined as volume stock, attribute <VolS:True> on the EPOSSProduct Collection object corresponding to product being transacted, then the attribute <VolSValue:> on the ModeParameters Collection object corresponding to the mode being undertaken, should be taken to direct the sale value according to the following table.

<VolSValue:> Value	Transaction Sale Value	Notes
Zero (literal "Zero")	Zero (0.00)	Minimum sale value check overridden
Sale	Product Sale Value as provided by EPOSSProducts <SV:> if fixed price or as entered if variable	Sale Value Range Check is not overridden
Loss	Product Loss Value as provided by EPOSSProducts <AP:> if present otherwise <SV:>, if fixed price or as entered if variable	Sale Value Range Check is not overridden For products with MultipleValue price restrictions, the Retail Price is substituted with the Adjustment Price and then a pseudo MultipleValue price is created based on the ratio of existing Multiple value to Retail Value times the Adjustment Price - after that point all checks are carried out as before.

There are migration implications of such a change. The change required must be coordinated with the office migrating to S80 and conducting accounting in Trading Periods rather than CAP. In other words it is only applicable to Stock Units after having rolled over from the final CAP into the first TP. Hence an additional check is required to ensure that the stock unit to which the user is connected has been migrated to using TPs. This is achieved by examining the stock unit <Collection:StockUnits> corresponding to the Stock Unit being used, and examining the attribute <TP:>. If <TP:> is set to 1 then the Stock Unit is operating in Trading Periods and so the changes can be effective.

FUJ00085124

5.1.4.4 Changes to Product Settlement

It has been a long standing cause of concern that Horizon settlement of transactions in modes other than serve customer is against products provided by POA rather than the Post Office. Historically these products have been of no interest to the Post Office. The introduction of Impact Release 3 presents the need to provide POL with details of at least some of these settlement transactions. As a result the opportunity arises to correct this situation.

As a result of the introduction of Impact release 3 the use of POA Products will be phased out with their replacement by Post Office defined settlement products.

The Type C Reference Data Collection <Collection:ModeParameters> will be updated for each object instance where there is a Settlement Product defined, attribute <SettlementProduct:>. The <SettlementProduct:> attribute value will be replaced by a Post Office Defined Product specified in the Type A Reference Data Collection <Collection:EPOSSProducts>.

This change can occur at any time and provided that the new Settlement Product has been made available as Type A Ref Data before the ModeParameters Object is changed, then that is all that is required. In particular the counter may still be running on the S75 code set – No S80 code changes are required.

Additionally the Type A Reference Data Collection <Collection:ProductModes> [RDMCHLD] will be changed so that a settlement product may be defined for each mode in which that product is transacted, attribute:<Mode:<S:>>. This replaces the S60 solution which had such overrides in a separate collection.

Changes are required to EPOSSCore and EPOSSSettlement, to implement changes to transaction settlement in order to utilise the changed settlement product reference data.

On settling a transaction if there exists a settlement product for the mode which is different from the default <SettlementProduct:> defined in the reference data collection <Collection:ModeParameters> then the transaction for the transaction product will be settled against the defined product.

There is no absolute need to tally such a change with the introduction of trading periods or managing stock by volume.

The primary mappings associated with the new products must be the same as those for the existing settlement products, otherwise balancing will be invalidated.

5.1.4.5 Changes to APS

As a result of the changes to transaction attribute structures arising out of the introduction of volume stock changes are required to APS to ensure transaction attribute content and format integrity is retained.

APS will be changed to generate transactions by a call to EPOSS Core, rather than creating the transactions itself. EPOSSCore will set up the transactions ready for committal but will not write the transactions to the message store.

Following the transactions generated by EPOSS Core post processing will then be undertaken by APS itself to ensure the complete integrity of APS transactions is attained.

The following post processing is required:

- Digitally sign the transactions as current

FUJ00085124

- Split Large transactions, Quantum transactions where a payload of data is carried which would otherwise cause the transaction to exceed the maximum size for a Riposte Message
- Add a Help Button Text and Caption, used to provide bespoke text displayed when an attempt is made to bin or edit an APS transaction on the stack

5.1.4.6 Merging Volume and non-value stock

The DP promotes the merging of volume stock (that stock which was volume and which is now managed by volume) with non-value stock.

At this time however there has been no agreement to undertake such a merger and hence no action is required at this stage.

5.1.5 Introduction of Volume Stock Rollover Data Model

The handling of Stock by Volume and the removal of the Cash Account both require changes to the recording of Figures at stock unit and office rollover. These figures provide the next accounting period, whether, BP, CAP or now TP, with the opening position in order that reports in the next accounting period can report the current stock position and remain in balance.

The following changes summarise the changes to opening figures production:

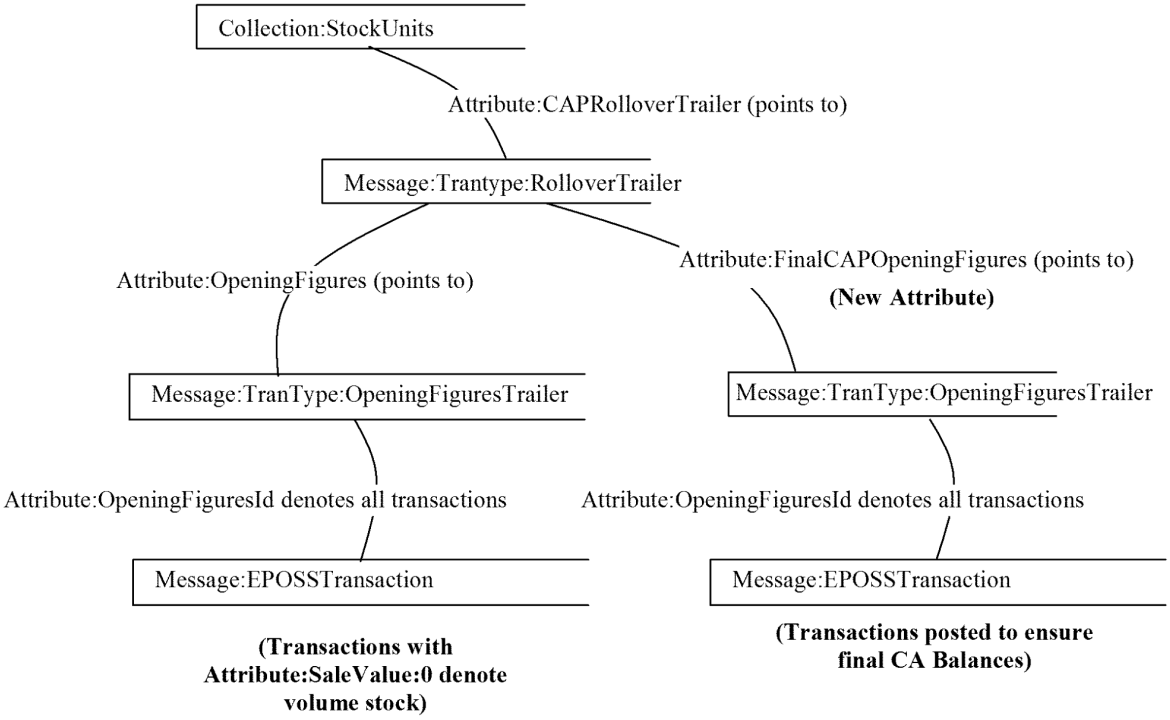
- Opening Figures for each stock unit will comprise figures recording stock managed by volume and figures recording stock managed by value.
- Office rollover figures produced for use exclusively for cash account production will no longer be produced. Otherwise Office rollover figures remain unchanged.
- Two sets of opening figures for each stock unit will be produced in the final CAP, before rolling into the first TP, the first set provides the opening position to manage stock managed by volume as above, but a second set of figures will also be produced as at present to support production of the final cash account.

5.1.5.1 Stock Unit Rollover Data Model

The stock unit Volume Stock rollover Data Model, as a consequence of the introduction of Volume Stock is largely unchanged, but for detail. The changes are limited to the introduction of additional attributes and changes to the population of existing attributes.

The following diagram represents the model.

FUJ00085124



5.1.5.2 Producing Stock Unit Rollover Figures

The changes required to implement the above model will be made to EPOSSStock Unit.

5.1.5.2.1 Steady State Rollover Processing

Steady State rollover processing is required as follows.

On rolling over a stock unit, opening figures will be produced largely as at present. A set of EPOSSTransactions will be produced, each transaction identified by the same value attribute:OpeningFiguresID, being the first of the set of records produced.

The set of transactions provides the stock product total position for each item of inventory stock, whether managed by value or volume. Those transactions with a sale value of zero denote stock managed by volume.

Transactions written with a sale value of zero are identified in the EPOSSProducts Collection by <I:True> and <VolS:True>.

The basis of the transactions will have already been produced by a DataServer build of the transaction hierarchy.

All transactions will be extracted from the node structure defined as Value Stock and MOP, node 3008. However the accumulator used to populate the transaction will vary depending whether the product is value or volume stock.

Value Stock Products will have the transactions populated from the accumulators sv and qty. Volume Stock Products will have the transactions populated from the accumulator qty, but with salevalue assigned zero. Determination of the product being value or volume stock is achieved by examining the <VolS:> attribute within the transaction.

FUJ00085124

These changes should only be actioned if the stock unit is rolling to another trading period, or a BP within a trading period.

5.1.5.2.2 Final CAP Rollover Processing for Stock Unit

In addition to the steady state processing changes a set of rollover figures will also be required to be produced when rolling the stock unit from the Final CAP into the first TP.

These figures will be produced following the processing rules of the current CAP rollover process, in other words providing closing stock value figures for each inventory item whether it is now defined by value or volume.

5.1.5.3 Office Rollover Data

Office Rollover figures recorded in order to produce the cash account (Cash Account Transactions) or representing the cash account (CashAccLines) will be removed. Their production entirely relates to production of the cash account.

Cash Account Transactions are identified as transactions (<EPOSSTransaction:> present) with a <PM:> also present pointing to a <L7:6999> node.

CashAccLines are identified as transactions (<EPOSSTransaction:> present) with a <CashAccLine:> attribute present.

These figures will be removed only after the last cash account and office rollover have completed.

5.1.6 Introduction of Trading Period Rollover Script

As a consequence of the introduction of the changes for Impact release 3 there are significant changes to the User Interfaces on the Counter. These are specified in references [DCRUI], [RPUI], and specifically [BTSUI]. It will be necessary to refer to these documents in association with this HLD.

This section defines the changes required to the stock and office balance processes as a direct consequence of the changes to the UI defined in [BTSUI].

The changes affect the following existing functions:

- Initiation of a Stock Unit Snapshot
- Initiation of Stock Unit Balancing
- Initiation of an Office Snapshot
- Initiation of Office Balancing

The section is divided into the changes which affect Stock Unit Balancing and those that affect Office Balancing.

All changes described affect EPOSS Stock Unit.

5.1.6.1 Stock Unit Balancing

5.1.6.1.1 Stock Unit Snapshot

FUJ00085124

A new report will be introduced into Global Objects to support the report layout for balance reports produced when implementing the requirements of Impact Release 3. The layout is specified in [RPUI].

A stock unit balance snapshot is initiated from a menu button on the Stock Balancing Menu. A different menu will be displayed depending whether the stock unit has already rolled to a TP or is still operating using a CAP. The button definition will be different for each menu, identifying a different report identifier depending which version of the report is to be produced.

Once both the report and the button are defined, there are no changes required to produce the appropriate snapshot.

5.1.6.1.2 Stock Unit Rollover

The dialogue undertaken to carry out stock unit balancing is changed under Impact Release 3. Whilst not a significant change, tablets and mnemonics are different and require to be managed correctly to ensure the correct dialogue is conducted whether a stock unit is carrying out a rollover within a CAP or a TP.

The rollover process will continue to carry out a number of sanity checks before allowing the stock unit to roll over. These will continue with some change. If the stock unit is already in a TP then:

- The check for outstanding transfers remains unchanged
- The check for mandatory reports remains unchanged
- The check for outstanding declarations remains unchanged, however the need to declare non-value stock will be removed and the function to support the declaration of non-value stock will be removed
- The check for outstanding discrepancies remains unchanged.
- The check on negative stock will be amended so that all inventory items, value and volume stock, are checked to ensure there is no negative volume within the stock unit

If this is the last stock unit to be rolled, then:

- All transaction corrections must have been applied (see 5.1.10). The check will be applied prior to the test for Parcel Traffic

Once the checks have been performed the rollover proceeds as at present.

The Stock Unit Balance Report will be specified in a new report definition. Initiated from the new stock unit balancing process the report will only be produced if the stock unit is rolling within a TP.

After producing the trial stock balance report, the option to rollover into a new BP or TP will result in further checks, before the rollover can be committed. The checks are only applied if the user chooses to roll forward into a TP:

- If there is a nett discrepancy, whether over or short, then the nett amount is moved to local suspense. The discrepancy amount is contra'd and a transaction recorded against the corresponding local suspense product

FUJ00085124

- If this is the last active stock unit to be rolled then the rollover may only be completed if there is no nett amount in Local Suspense for the entire branch. If there is a nett amount in local suspense the user is prompted to clear that amount (see 5.1.12)

5.1.6.2 Office Balancing

The Office Balancing Process remains largely the same, but for the replacement of the Cash Account by the Branch Trading Statement.

A new Office Balancing Menu is specified which will be visible only if the Branch is balancing a Trading Period, as opposed to a CAP. In other words all stock units have rolled to a TP.

Whilst the appearance of the menu is different the underlying functions remain largely unchanged. The following Button changes are evident:

- The button to initiate Office Balancing will be changed to 'Trading Statement'. The processing initiated will be the same
- The button to denote the Office CAP will be changed to 'Branch TP'. The processing initiated will be the same

The Office Snapshot button remains unchanged.

5.1.6.1.1 Office Snapshot

A new report will be introduced into Global Objects to support the report layout for balance reports produced when implementing the requirements of Impact Release 3. The layout is specified in [RPUI].

An office balance snapshot is initiated from a menu button on the Office Balancing Menu. The button definition will be different for depending which office balancing menu is specified, identifying a different report identifier depending which menu is being specified.

Once both the report and the button are defined, there are no changes required to produce the appropriate snapshot.

5.1.6.1.2 Branch TP

The processing to determine the current TP remains unchanged, the value being held in the Office Persistent Object. However the tablet displayed requires change to encapsulate the office operating Trading Periods rather than CAPs. The code must be changed to check the status of the office and display the 'accounting period' together with the appropriate tablet.

5.1.6.1.3 Office Balancing

The existing sanity checks employed to determine whether office balancing may continue, are retained. However the displayed tablets in the event of the checks reporting an error or warning require change to adopt the convention for TPs rather than CAPs.

The existing check for the entry of non-value stock having been made will be removed.

The production of the cash account will no longer be employed if the Office is operating in a TP. Instead an invocation of BESReports will be made to print the Branch Trading Statement. Both Trial and Final versions of the report exist replacing directly the trial and final versions of the Cash Account. Unlike the cash account only a single copy of the final BTS is produced.

FUJ00085124

5.1.7 Introduction of Volume Stock Balancing Model

The implementation of Volume Stock, introduction of a local suspense account and the changes to the processing of declarations and variances for Impact Release 3 requires changes to the balancing model that underpins stock unit and office balancing.

The balancing model is supported by reference data and in particular the reference data defining the EPOSS Accounting Node Hierarchy.

This section specifies the changes to the balancing model, implemented as reference data, in order to ensure the counter balancing process is capable of balancing the stock unit and office. The changes provide rules and constraints that must be adhered to in order that Stock unit rollover balancing will have Payments equal to Receipts.

In applying these changes it is imperative, so as to retain a balanced view of transactions, that only node additions are made, or attribute (accumulator) additions are made. Removal of an accumulator or removal of a node may cause the balance view to be corrupted.

These are reference data only changes, that impact the delivery of type A and B data and are hence involve the participation of POL.

5.1.7.1 Tertiary Mapping Nodes

The concept of Tertiary Mapping has already been introduced. Reference Data is required to support the required Tertiary Mappings.

Each stock product that is defined as being managed by volume will have a Tertiary Mapping defined. The node hierarchy is required to support all Tertiary Mappings.

Any tertiary mapping is a mapping of a volume stock product into the Receipts or Payments table and provides the balance report analysis structure for the sale or adjustment of such products.

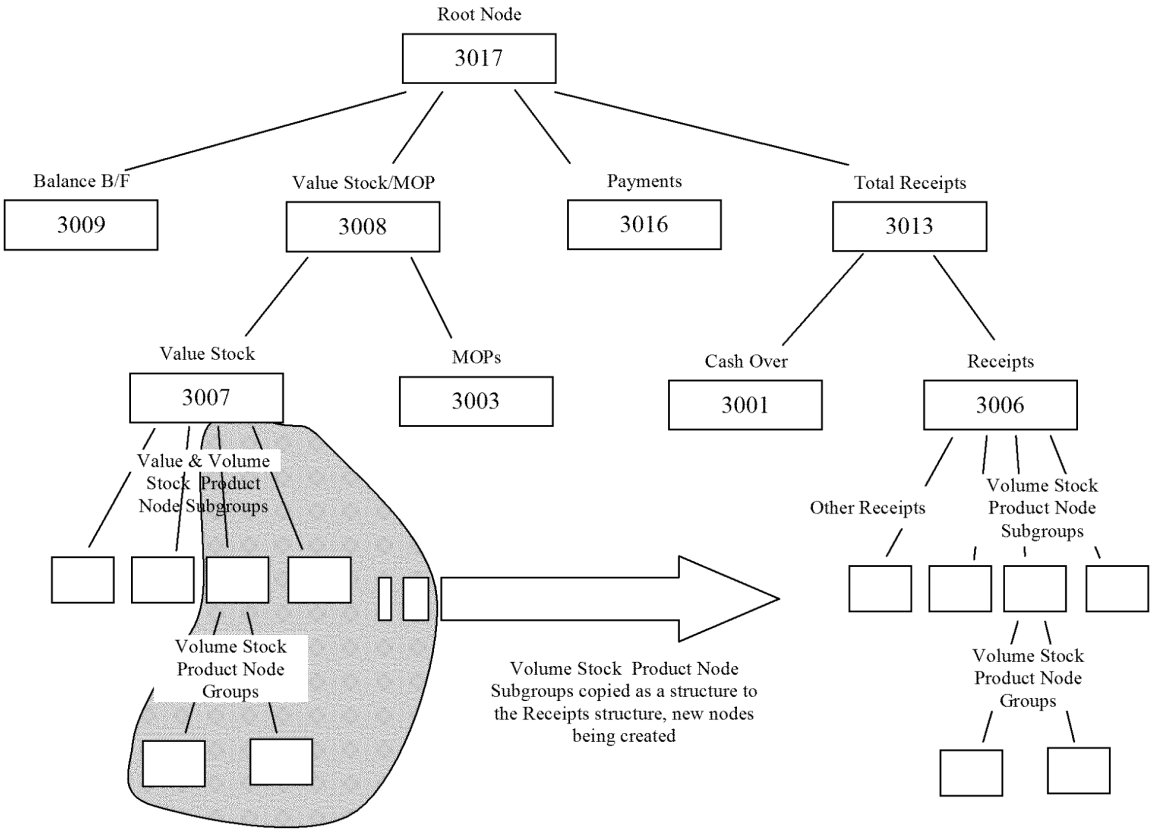
In reality there will unlikely be a mapping to the payments table as the sale of stock like a receipt product constitutes the giving of a product in return for method of payment. Unlike a payment which is the giving of method of payment in return for product.

The tertiary mapping structure has to define a structure for all those products currently managed by value, and having primary mappings that report to the value stock table, but will now also have a tertiary mapping into the Receipts Table.

It is anticipated that the structure will be a clone image of that that exists for primary mappings. However POL together with RDT may adopt and implement a variation which satisfies the analysis requirements of transacting volume stock, within the constraints that follow.

The following diagram represents those elements of the Accounting Node Hierarchy which are impacted by change. The diagram does not represent the full extent of the hierarchy, only that impacted by change.

FUJ00085124



The only dependency is that the node structure must map into the level 3 node 3006.

The primary mapping structure for these products must remain intact as it will support the figures to analyse the stock quantity figures for all volume stock.

5.1.7.2 Applying New Accumulators

The addition of the tertiary mapping nodes supports the mapping of volume stock sales to receipts.

Section 5.1.3.2.1 introduces the concept of new accumulators. The use of these accumulators is to allow primary mapping structures to support the accumulation of figures for both value stock sales and quantities, and volume stock quantities.

Four new accumulators have been identified in section 5.1.3.3.1.

Essentially operating in pairs, they:

- Allow the aggregation of sale value and quantity for transactions into the primary mapping stock structure, for products that **do** have a tertiary mapping defined, ie. volume stock products. They are SVETM and QTYETM.
- Allow the aggregation of sale value and quantity for transactions into the primary mapping stock structure, for products that do **not** have a tertiary mapping defined, ie. value stock products. They are SVNTM and QTYNTM.

FUJ00085124

- Count transactions into the primary mapping stock structure for products that do/do not have a tertiary mapping defined – i.e. volume/value stock. They are counted into RCETM/RCNTM respectively

These new accumulators only apply to a product defined as an Inventory Product, volume or value stock.

The following changes to defined nodes are required.

All nodes defined in the node structure which support volume stock products, <I:True> and <VolS:True>, will have the additional accumulator QTYETM defined. There is no need to also define SVETM as there is no record of the value of volume stock movement, though any future instances will be supported without software change being needed. The addition of this accumulator should be defined on all nodes in the structure up to 3007. This allows the net stock quantity holding as a result of all movements of volume stock to be aggregated. Support for an aggregate quantity has limited meaning, but is supported to comply with existing accumulations.

All nodes defined in the node structure which support value stock products, <I:True> and <VolS:False> will have the additional accumulators defined SVNTM and QTYNTM. This allows the net stock value and quantity of all value stock movements to be aggregated. It is necessary to apply these changes so making the use of sv and qty accumulators redundant. Failure to apply these changes will mean that volume stock value and quantity figures will corrupt the overall value stock value and quantity figures. Whilst quantity totalling has little relevance at the highest level, eg. the total quantity of volume stock, it does have relevance on lower subtotals. Equally whilst some volume stock movement does not have a sale value other movements do and hence will corrupt the value stock highest level totals.

At this juncture it is possible that such accumulators may be avoided. Ie. SVETM, sv & qty do not apply. Also, QTYETM & QTYNTM may not be required at the higher levels of accumulation.

All nodes defined in the node structure which support value stock products, <I:True> and <VolS:False> will have the additional accumulators defined RCNTM and RCETM. This allows the number of volume/value stock movements to be counted separately in order that grouping lines on the balance report can be suppressed or not according to whether value/volume stock has been transacted.

These changes must be applied before S80 code is activated but otherwise can be applied at any juncture. At Point 20 S80 code will be run and this will have knowledge of the new accumulators but as the reports to print them will not change until Point 50, the Point 20 code will be emulating S75. There is no migration issue on removal of the unused accumulators as they are not reported on.

5.1.8 Introduction of Stock Balance Reports

There are substantial changes to the format of the Balance Reports produced for each stock unit and the office as a result of the changes for Impact Release 3. The content of the reports for stock unit balance snapshot, trial and final, plus the office balance snapshot, trial and final are very similar, and the changes applicable are equally applied similarly.

The specifications of each report are provided in [RPUI] and [REPREC]. The following sample is provided to highlight the changes.

FUJ00085124

New report definitions are required in Global Objects to support these reports. It will not be possible to amend the existing definitions. Therefore under the new scheme if the stock unit is operating as a TP then the new report layout will be used, otherwise the old will be used. When moving from CAP to TP the old one will be used (even if the new code is running).

Each report is driven by a dataserver aggregation build described in 5.1.3. Hence in order to define the changes required to the balance reports it is necessary to understand the structure of the nodes that support the format of the reports. The following table decomposes the above report into the constituent 'sections'.

Ref	Report Layout 1 2 3 4 12345678901234567890123456789012345678901234567890123456789012	Node	Acc.	Notes
01	Feltham Post Office FAD: 123456X			
02	11:42 17/01/1998 TP:01 BP:01 SU:SH1			See 5.1.2.7.2
03	Trial Balance - Office Copy			
04				
05	******Discrepancies in this Account******			
06	*Discrepancy OVER 20.00 *	3111	sv	
07	*Discrepancy SHORT 0.00 *	3112	sv	
08	* ----------*			
09	*Nett discrepancy 20.00 *	3110	sv	
10	* ----------*			
11	*Excess Cash Removed 0.00 *	971	sv	
12	*Cash Shortage Made Good 0.00 *	972	sv	
13	* ----------*			
14	*Nett Cash Adjustment 0.00 *	970	sv	
15	* ----------*			
16	**********			
17				
18	VALUE ITEMS & MOP VOLUME VALUE			
19				
20	Cash 118.60	3003	qty/sv	
21	Cash 118.60	..	sv	
22	Cheque 116.00	..	qty/sv	
23	Cheques 116.00	..	sv	
24	MOP 234.60	..	sv	
25	Euros 590.00	3007	qty/svntm	
26	Fgn Currency Sterling Equ 590.00	..	svntm	
27	BUREAU DE CHANGE 590.00	..	svntm	
28	Postage stmp 10.00	..	qty/svntm	
29	Other Postage Items 10.00	..	svntm	
30	POSTAGE 10.00	..	svntm	
31	----------			
32	TOTAL VALUE ITEMS & MOP 834.60			Note 1
33	----------			
34				
35				
36				
37				
38	RECEIPTS VOLUME VALUE			
39				
40	Balance B/Fwd 0.00	3009	sv	
41				
42	Colour TV Non AP 2 232.00	3006	qty/sv	
43				

FUJ00085124

44	Currency Reval Up	1	20.00			
45	B de Change-Revalue Up		20.00			
46	Sale First Class Stmps	10	2.60	3006	qty/sv	
47	Sale of Stock		2.60	3006	qty/sv	
48	Transfers In		0.00	3033	sv	
49						
50						
51	Rem In Supp Div		0.00			
52	Rem In Other Pos		0.00			
53	Rem In Client		0.00			
54	Rem In Auto Dist		610.00	3028	sv	
55	REMITTANCES IN		610.00	3028	sv	
56						
57	Discrepancy SHORT Transferred		0.00	963	sv	
58	Discrepancy SHORT Resolved		0.00	964	sv	
59			---------			
60	TOTAL RECEIPTS		854.60			Note 2
61			---------			
62						
63	PAYMENTS	VOLUME	VALUE			
64						
65	Curr Reval Down	1	30.00			
66	B De Change- Revalue Dn		30.00			
67						
68	NatLot Prize	2	20.00	3005	qty/sv	
69	LOTTERY PAYMENTS		20.00	3005	qty/sv	
70	Transfers Out		0.00	3034	sv	
71						
72	Rem Out Supp Div		0.00			
73	Rem Out Other Pos		0.00			
74	Rem Out Data Cen		0.00			
75	Rem Out Client		0.00			
76	Rem Out Auto Dist		0.00	3029	sv	
77	REMITTANCES OUT		0.00	3029	sv	
78						
79	Discrepancy OVER Transferred		0.00	965	sv	
80	Discrepancy SHORT Transferred		0.00	966	sv	
81	Total VALUE ITEMS & MOP		834.60			Note 3
82						
83	Nett discrepancies		20.00-			Note 4
84						
85			---------			
86	TOTAL PAYMENTS		864.60			Note 5
87			---------			
88						
89	Balance C/Fwd		834.60	3008	sv	
90						
91	STOCK VOLUMES	VOLUME				
92						
93	1st Class Stamps	90		3007	qtyetm	
94	POSTAGE			3007	qtyetm	
95						
96	EXAMINATION					
97	Drawer examined and cash and stock found					
98	as shown in this summary					
99			Datestamp			
100			+-------+			
101	Signature		. .			
102			. .			

FUJ00085124


```
103                                    .       .
104  Time . . . . . .AM/PM  +-------+
105
106
107  TRANSFER
108  Cash and stock in this summary have been
109  transferred to me
110                                  Datestamp
111                                  +-------+
112  Signature . . . . . .  .       .
113                         .       .
114                         .       .
115  Time . . . . . .AM/PM  +-------+
116
117
118           *** END OF REPORT ***
                1         2         3         4
       12345678901234567890123456789012345678901234567890
```

In the above report, most stock under node 3007 has been converted from value to volume accounting, so is no longer reported in the 'Value Items and MOP' section. However, some items are retained as Value Stock. In particular products such as 'Other Stamps' under the 'Other Postage Items' node 1707. As a result there is a need to conditionally suppress reporting of the nodes under 3007 according to whether there have been any products transacted under the node which are still value stock.

The group and subgroup lines in the value stock section are therefore suppressed if the associated nodes have zero RCNTM accumulator.

In principle, the group and subgroup line in the volume stock section must similarly be suppressed if the associated nodes have zero RCETM accumulator. However, it has since been decided (PEAK 117170) that group and subgroup lines should always be suppressed for the volume stock section.

The following table provides details of those report lines that are arrived at by a combination of nodes identified above by Note references.

Note	Description	Node Combination
1	Total of Value Stock and MOPs	3003(sv)+3007(svntm)
2	Total Receipts	3013(sv)+3009(sv)+3028(sv)+3033(sv)+3011(sv)
3	Total Value Stock & MOPs	3003(sv)+3007(svntm)
4	Nett Discrepancies	3110(sv)
5	Total Payments	3016(sv)+3008(svntm)+3029(sv)+3034(sv)+3015(sv)

A change is required to the reporting component EPOSSReportProcessor to implement the above changes to combination nodes. The current component does not allow different accumulators to be applied when calculating the result of a combination node equation. A change is required to the component to allow this circumstance for Impact Release 3.

FUJ00085124

5.1.9 Introduction of Branch Trading Statement

A new report, the Branch Trading Statement is introduced as part of Impact release 3 to replace the Cash Account.

Production of the cash account will be inhibited and replaced by the branch trading statement, once the office has rolled into the first trading period, and thereafter. Unlike the Cash Account the BTS has no electronic interface to POL of the Reports' contents.

The specifications of the branch trading statement are provided in [BTSUI] and [REPREC], both documents providing reference information that supports the design solution for production of the report.

A new report will be developed in BESReports to implement the BTS. The report layout will be hard coded, i.e. is not driven by Global Objects, however the design solution provides for the data derivations being mostly soft, i.e. supported by Reference Data definitions.

The report is defined as two potentially multi-page sections. The first section reports the final Value movements and actions each Stock Unit, Suspense and the branch in the last TP, and is referred to as the 'Branch Trading Statement Summary Page'. The second section reports the final Volume Stock position for the branch at the end of the TP, together with the number of Transaction Corrections conducted, and is referred to as the 'Branch Trading Statement Stock Holdings'.

The BTS is an A4 landscape report produced in 'Trial' and 'Final' versions, and is support by the requirement to preview. A reprint facility is available, specified in Section 5.1.11.

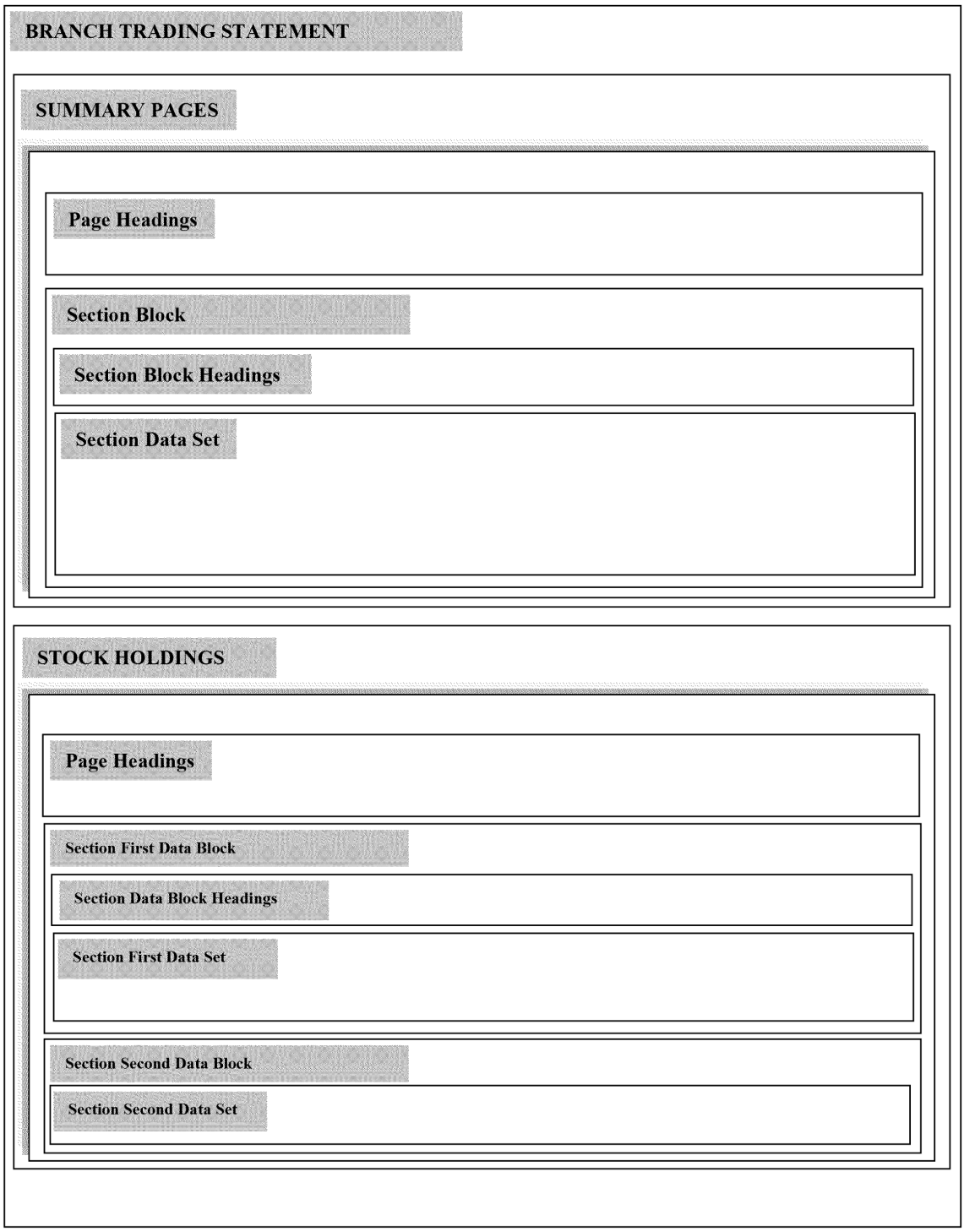
To produce the desired report, the structure of the solution within BESReports requires two phases, each handling a section of the report. The following paragraphs describe the design solution for the BTS, supporting the concept of the report in two sections, the following diagram depicting the schematic representation of the BTS in two sections.

The Summary Page consists of a single Block of data headed by a row of column headings. The Section Data Set specification is provided in the following sections.

The Stock Holdings Page consists of two Blocks of data, the first headed by a row of column headings, and the second providing final data to be printed on the report together with report termination information.

FUJ00085124

5.1.9.1 BTS Schematic Layout



Management of the Report Layout will be entirely under the control of the BESReports Code. Control of page throws and the required page headings will be driven and populated internally, though parameters will be provided in order to denote 'Trial' and 'Final' versions of the report.

FUJ00085124

Ideally it is hoped that all the lines for the summary page can be contained within a single page, so long as the number of stock units does not exceed four, through optimum use of Font sizes.

The number of pages constituting the Stock Holdings section is dictated by the number of separate Volume Stock Items transacted through the Trading Period and held at the branch.

5.1.9.2 BTS Summary Pages

The first section of the report provides a summary of Stock Unit, Suspense and Branch Value movements and actions applicable to the TP being rolled over at the branch.

The Section is formulated, over and above the page headings, in a tabular form as a set of rows and columns depicted as the Section Block:

- Section Block Headings
- Section Data Set
    - Row Headings Column
    - Branch Totals Column, which equates to the total of all column entries for the same row
    - Suspense Entries Column
    - Stock Unit Entries Column, there being one column per stock unit

The number of pages required to populate the section block will depend upon the number of Stock Units. Should there be more than four stock units then more than one page will be required. The Section Block Headings will be determined by it being the first or subsequent page, as branch and suspense detail is not repeated on subsequent pages and hence the Section Block Headings will change also.

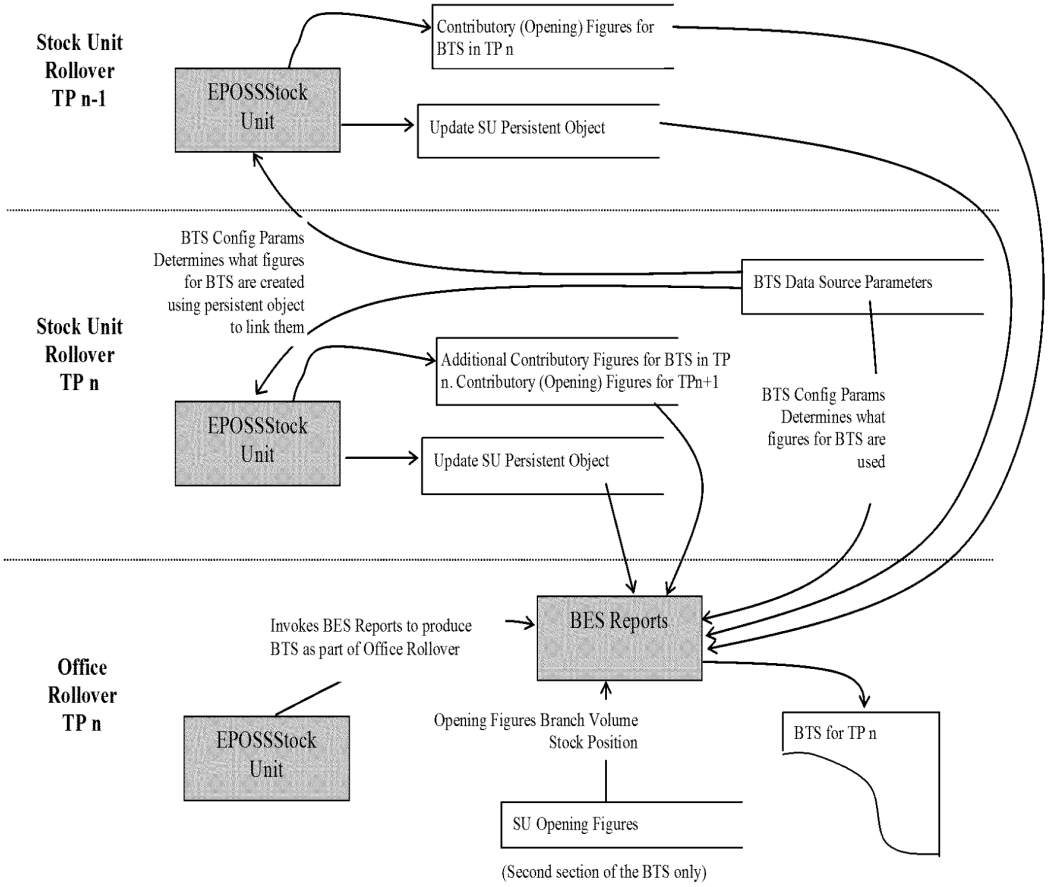
The production of the Section Data Set is implemented as a soft driven solution driven by a set of parameters, known as the BTS Data Source Parameters in reference data, See Appendix B <Collection:BTSSourceData>. The parameters are conceived by recognising that the data set is essentially a set of rows, and each row having a particular definition for the stock unit or suspense item being reported on. Branch totals themselves are only a sum of all columns on the row, and are not driven by data. BESReports will arrive at the Branch Totals from the contributing Stock Unit and Suspense row items.

The parameters also allow the source data, which is complex and derived from a multiplicity of sources, to be gathered as part of the balancing process, when it is readily available, and made available to BTS production in an orderly manner.

The following diagram provides a schematic of the section data set data gathering and presentation process.

The reference data provided in BTS Data Source Parameters provides the key to the solution.

FUJ00085124



The data provides a mechanism to instruct various items of data, needed to populate the BTS, to be written out at the point the stock unit (or office in the case of some suspense items) is balanced, importantly, in a format that simply provides BESReports, with a series of messages allowing the report to be produced simply. The following table depicts the schematic of the data.

Data Item provided reference data	Description
Line Identifier	The data recognises that the BTS summary page data is a series of rows, encapsulating data and headings. New lines may need to be added, or removed. Each line is therefore given an identifier. These identifiers will form the object names to the reference data collection. An example would be LineCCC. The considered alphabetic suffix allows lines to be removed or added without problems had the lines been numbered consecutively
Row Heading	This is the specific text that leads the row. The data recognises blank lines are needed and allows 'White Space' as a heading to depict a blank heading. Without any further instruction BESReports will produce a blank line if a row heading of white space is encountered.
Accounting Period (AP)	Data sourcing a BTS instance is derived from both the TP being rolled, and from the previous TP. The AP (Accounting Period) denotes whether the row data is derived from the current (C) or previous (C-1) TP.

FUJ00085124

PrintMethod	A given row element can have a specific and defined number of derivations. It can be a total of other elements on the report. It can be white space, or it can be a data item. PrintMethod defines a 'Method' known to BESReports which provides an instruction as to how to derive the line element. If no PrintMethod is defined then this denotes a new line which is being introduced but is not yet to be printed. In other words the data gathering stage has started but print implementation has yet to commence. There are two defined methods currently defined, LocalRetrieve and DataRetrieve. Local Retrieve identifies an instruction to amalgamate the content of other rows (for the same column) into this element. DataRetrieve identifies an instruction to go and retrieve the data element from a series of messages, identified by the Line identifier.
PrintMethod Attributes	Attributes used by the PrintMethod. The print attributes specify an 'equation' string to allow the content of the other line identifier cells and/or source values. The Accounting Period element value will also assist in data gathering.
SourceMethod	A given row element can have a specific and defined number of derivations. Most commonly it is an item of data derived from the TP rollover position for an SU or Suspense. Such data items are arrived at from different data sources, For example it may be the stock in hand value of cash, established from the SU rollover figures or it could be the total amount of receipts for the TP, less Remittances and Transfers. SourceMethod defines a 'Method' known to EPOSSStockUnit which provides an instruction to derive the line element at Stock Balancing and store the data item in readiness for BTS production in a form for easy retrieval. A new line on the BTS could be introduced as a future change by defining new lines and a Source method without a print method. The data will be prepared and stored in readiness for printing on BTS instances in the future. There are two methods currently defined, SumAcc, which identifies an instruction to amalgamate the content of Accumulated Nodes in the Accounting Node Hierarchy, and SumSusp which identifies an instruction to amalgamate the content of office suspense
SourceMethod Attributes	Attributes used by the SourceMethod. These specify an 'equation' string to allow the content of the node and product accumulations to be calculated. The Accounting Period element value will also assist in data gathering.

5.1.9.2.1 Stock Unit Rollover

As part of stock unit balancing EPOSS Stock Unit will change to encompass changes to soft write additional rollover data to be used at the point of BTS production. The changes are accompanied by enhancements to the Stock unit rollover data model to allow BESReports to have knowledge of the navigation to the data to be used for the BTS.

Two areas of change are required:

- New rollover trailers are to be introduced which allow EPOSS Stock unit to record BTS data against. The same rollover trailers become known to BESReports allowing the data to be retrieved
- A reference data mechanism is introduced which allows, under data control, EPOSS StockUnit to record data items for printing on the BTS at rollover time. The same mechanism allows BESReports to retrieve those items for printing

FUJ00085124

The BTS Summary Page essentially contains information derived from the previous CAP/TP (Brought Forward Figures into the current TP), and the current TP (movements within the TP and the Carried Forward Figures to the next TP).

The strategy being adopted for the BTS is to limit the amount of data retrieval navigation work it has to do and maximise the ability for the figures to be provided to the BTS by the Rollover process, when the source data is readily available and can be simply amalgamated into a form for the BTS.

To implement the fact that a BTS Report contains data from the previous TP and the current TP the basic premise is that the rollover process, both stock unit and office, will record data at TP rollover, for the BTS of the current TP and the next TP, the latter being its brought forward figures.

Driving the data to be reported in a soft manner, the 'system' makes no assumption that the brought forward figures to be reported are the same as the carried forward figures reported from the previous TP, hence allowing for reconciliation if so desired and avoiding more knowledge about the report content than is necessary.

For stock unit rollover a system of new rollover trailers support the ability to provide the storage of data for the BTS. Three new rollover trailers are defined.

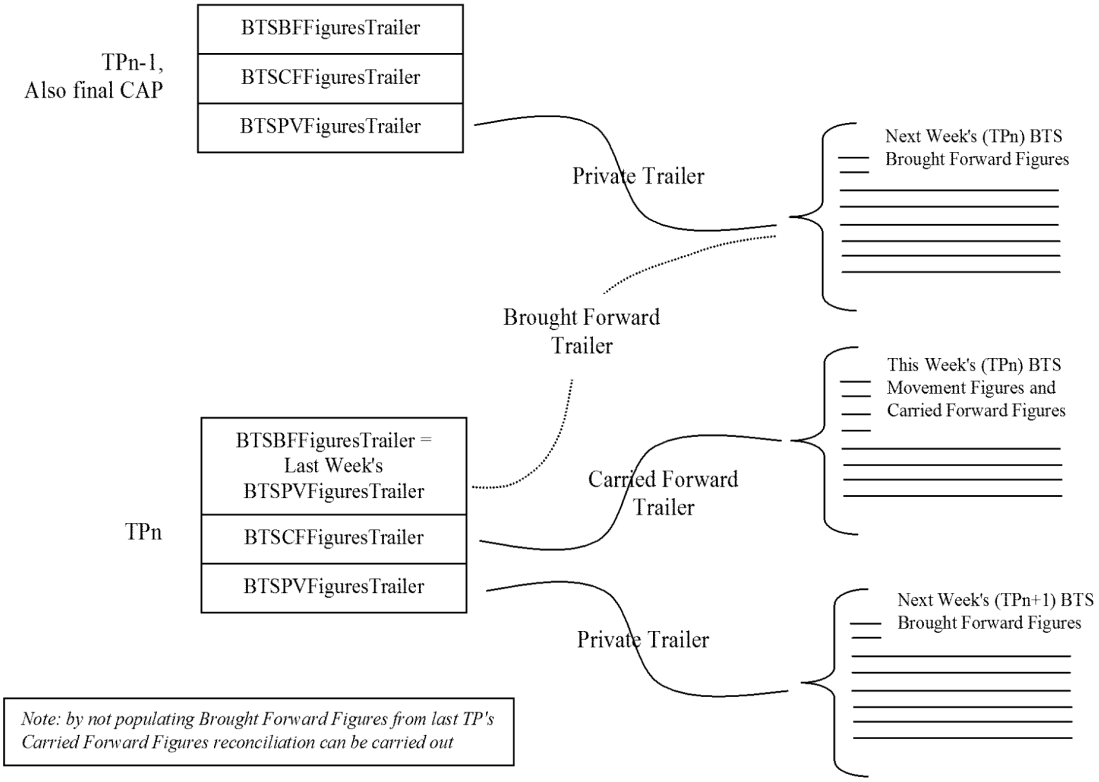
Each new Rollover Trailer specifies the Opening Figures Message Identifier that groups the data messages recorded for later use by the BTS. The rollover trailer itself is defined as by <TranType:>.

Each new rollover trailer is pointed to from the Stock Unit Rollover Trailer <TranType:RolloverTrailer>, specifying the Message identifier of the rollover trailer, which in turn is pointed to by the Stock Unit persistent object, both part of the existing rollover process model, as described in section 5.1.5.

Responsibility rests with EPOSS StockUnit to manage and maintain the rollover trailers, and the data messages linked to each trailer. BESReports has the responsibility to use the data in BTS production. It does not manage or maintain their existence, though it will police any errors in their presence being missing.

Processing of the rollover trailers is depicted in the following diagram, the mechanism providing the capability to supply all stock unit related figures on the BTS.

FUJ00085124



5.1.9.2.2 Office Rollover

Stock Unit rollover figures are sufficient to provide the data population for all Stock Unit related figures. Problems occur however in using these figures in aggregated form, to populate suspense items on the BTS. To resolve this Office level figures are also exposed by EPOSS Stock Unit to provide data to the BTS, in the form of ## figures.

In so doing consideration is given to the fact that the same figures should be capable of being used for Suspense Account Reporting also.

The concept of ## figures exists for Cash Account Production, but can also be used to provide office level data for the BTS. The removal of the cash account means that the ## figures can also be adapted to suit the BTS as there is no need to support the cash account any longer.

A similar principle of new rollover trailers will be applied to office rollover as is the case for stock unit rollover. However the navigation mechanism will be slightly different and the trailer will only support Brought Forward Figures. The new rollover trailers will be identified by the same naming convention, utilising the BTS Source Data parameter Object.

This allows EPOSS Stock Unit to record office level data items, currently only Suspense items, so that they can be picked up and printed by BESReports on the BTS.

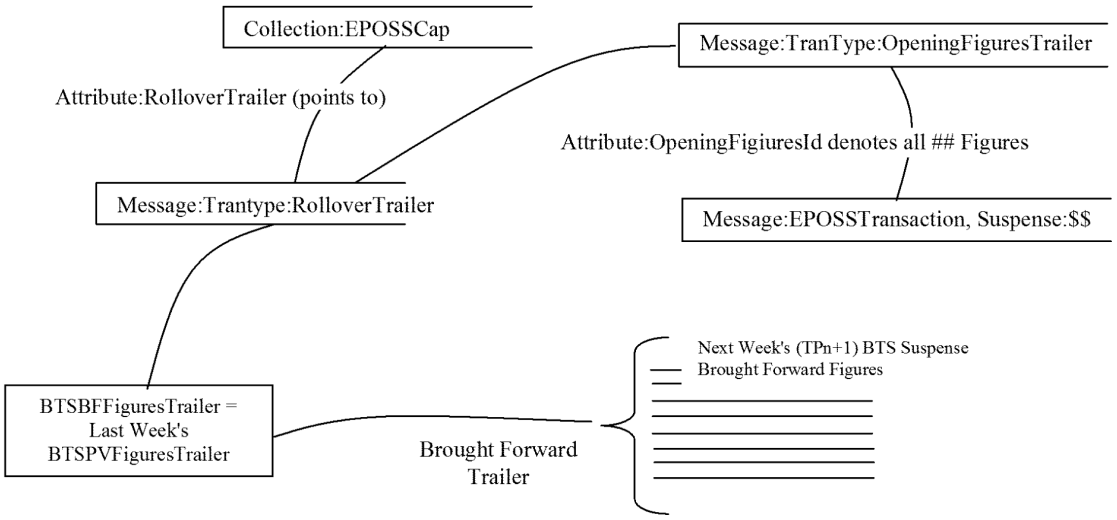
Hence the following data can be derived using the office rollover figures:

- Brought Forward Suspense Figures
- Carried Forward Suspense Figures

The principle can be used to support the brought forward figures to the first BTS.

FUJ00085124

Access to the rollover trailers is achieved by the following scheme.



5.1.9.2.3 Soft Data Control

The new rollover trailers provide the structure around which EPOSS StockUnit and BESReports will communicate the data to be provided for, and reported on the BTS.

EPSOSS StockUnit will use the reference data collection BTS Source Data as the driver to the data written. As part of the balancing process, and with the Data Server tree built for the current TP rollover, EPOSS StockUnit will scan each object instance of the table. For each entry with a defined SourceMethod, the source method will be executed. All SourceMethods assume the Data Server tree has been built and is available.

The execution of a source message results in a Data Message being written to the message store recording the BTS Line identifier, the value to be printed and the rollover trailer message identifier with which all like messages are to be linked, based on <AP:>.

The BTS Source Data attribute <AP:> is provided within the BTS Source Data to link the data EPOSS Stock Unit processes to the appropriate rollover trailer. In processing (rolling over) TPn any BTS Source Data object denoting the current AP ('C') will have data linked to the BTSCFFiguresTrailer (Carried Forward Trailer). Any object denoting the current-1 AP ('C-1') is instructing EPOSSStockUnit to prepare figures for the next TP and hence data will be linked to the BTSPVFiguresTrailer (Private Trailer).

In the case of a report line object being specified as <AP:C> BESReports will be pointed to the BTSCFFiguresTrailer, and in the case of <AP:C-1>, will be pointed to the BTSBFFiguresTrailer.

As part of the generic stock unit rollover process the BTSBFFigureTrailer will be assigned the BTSPVFiguresTrailer identifier of the previous week's Private Trailer.

The BTS Source Data reference data collection <Collection:BTSSourceData> provides the objects that EPOSSStockUnit has to populate data for. In order to ensure BESReports and EPOSS StockUnit have a common interface to the data via the rollover trailers, the rollover trailer identifiers themselves are specified in reference data, through the <ObjectName:Parameters> instance of the collection.

FUJ00085124

The same object instance also provides BESReports with a set of PrintControl parameter specifications that can be varied.

All data specifications are provided in Appendix B.

5.1.9.2.4 BTS Summary Pages Derivation

The Summary Pages consist of a sequence of sections. The content of each of these is defined by a set of summary line definitions in <BTSSourceData>:-

- Brought Forward Figures (LineB*)
- Receipts (LineC*)
- Payments (LineD*)
- Carried Forward Figures (LineE*)
- Trading Position (LineF*)
- Local Suspense (LineG*)
- Make Good and Remove Excess Cash Events (LineH*)
- Branch Adjustments (LineI*)

The following tables indicate the data item derivations for those summary lines. The line/field names and layout are as described in [BTSUI]

FUJ00085124

BTS Section B – Brought Forward Figures

Cash on Hand B Fwd **LineBB**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Value derived from Opening Figures Node **1001** (Cash) plus **3110** (Discrepancies). Should match the corresponding **Cash on Hand C Fwd** (LineEE) in previous Branch Trading Statement. N.B. Discrepancies is normally zero (because any discrepancy must be transferred to Local Suspense before producing the Final Balance), but for the special case of the first TP, there may be a nett discrepancy carried over from the final CAP. In that case the **Cash on Hand B Fwd** figure will not agree with the balance report for the final CAP, but including the discrepancy here allows the overall Trading Position to remain zero as required by PEAK 118648 (See Section F).
Cash Awaiting Collection B Fwd **LineBBB**	Branch Total	Same as suspense column
	Suspense	Value derived from Non Inventory Opening Figures Products **5610** and **6509.** Should match the corresponding **Cash Awaiting Collection C Fwd** (LineEEE) in previous Branch Trading Statement.
Suspense B Fwd **LineBBBB**	Branch Total	Same as suspense column
	Suspense	Value derived from Non Inventory Opening Figures for Nodes **740**, **490** minus Products **5610, 6509.** Should match the corresponding **Suspense C Fwd** (LineEEEE) in previous Branch Trading Statement.
Other MOP B Fwd **LineBBBBB**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Value derived from Opening Figures Node **3003** (MOP) minus Node **1001** (Cash). Should match the corresponding **Other MOP C Fwd** (LineEEEEE) in the previous Branch Trading Statement.
ForEx B Fwd **LineBBBBBB**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Value derived from Opening Figures Node **2016** (Bureau de Change). Should match the corresponding **ForEx C Fwd** (LineEEEEEE) in the previous Branch Trading Statement.
Other Postage B Fwd **LineBBBBBBBB**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Value derived from Opening Figures Node **3007** (Value Stock) minus Node **2016** (Bureau de Change). Should match the corresponding **Other Postage C Fwd** (LineEEEEEEEE) in the previous Branch Trading Statement.

FUJ00085124

BTS Section C - Receipts

LineC	**Blank Line**	
Remittance In Total **LineCC**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Total cash, cheques, stamps, foreign exchange and other Value Stock remitted in during the Trading Period. Corresponds to the **REMITTANCES IN** total on the SU Final Balance report. Value derived from Non-inventory Figures accumulation of Node **3028**
Cash Rems from SUs **LineCCC**	Branch Total	Same as suspense column
	Suspense	The sum of cash transferred into pouches pending dispatch from the branch during the Trading Period. Corresponds to the **CASH REMS MOVED TO SUSPENSE** total on the Suspense report. Value derived from Non Inventory movements of Product **5610.** (Same as sum of **LineDDDDB** for all SUs).
Gains to/from Suspense **LineCCCC**	Branch Total	Total of individual values in suspense and stock unit columns
	Suspense	Total of all losses transferred from stock units to suspense during the period. Value derived from the sum of all the Stock Unit column entries for **Losses to/from Suspense** (LineDDDD)
	Stock Unit	Total nett value of gains transferred into suspense from the stock unit during the period. Corresponds to the **GAINS TO/FROM SUSPENSE** total on the SU Final Balance report. Value derived from Non Inventory movements within the TP under Node **490**
Cash Pouches Despatched **LineCCCCB**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	The sum of cash in pouches despatched from the branch via the Stock Unit during the period. Value derived from Non Inventory movements of Product **6509**
Transfers In from other SUs **LineCCCCC**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Transfers of cash, cheques, stamps, foreign currency and other Value Stock from other stock units. Corresponds to **Transfers In** Total on the SU Final Balance report. Value derived from Non-inventory Figures accumulation of node **3010**
Other Receipts **LineCCCCCC**	Branch Total	Total of individual values in stock unit columns

FUJ00085124

	Stock Unit	Total value of goods and services (including banking deposits) which have been sold during the Trading Period regardless of MOP used. Reversals are included with the appropriate sign, and stock adjustments are included (at loss value). Corresponds to **TOTAL RECEIPTS** on the SU Final Balance report less the total of the following stock unit fields on the Branch Trading Statement: **Remittances In Total** (LineCC) **Gains to/from Suspense** (LineCCCC) **Transfers in from other SUs** (LineCCCCC) **Total B/Fwd** (Section B) Value derived from Non-inventory Figures accumulation of nodes **3013** (Total Receipts), **963 (**Gains to Local Suspense) and **966** (Clear Loss from Local Suspense), minus **490** (Gains to/from Suspense)

FUJ00085124

BTS Section D – Payments

LineD	**Blank Line**	
Remittances Out Total **LineDD**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Total cash, cheques, stamps, foreign exchange and other Value Stock remitted out during the Trading Period. Corresponds to **REMITTANCES OUT** total on the SU Final Balance report. Value derived from Non-inventory Figures accumulation of node **3029**
Cash Despatched via SUs **LineDDD**	Branch Total	Same as suspense column
	Suspense	The sum of cash collected from the branch over the period. Corresponds to the total of **Despatched Cash in Pouches** during the Trading Period on the Office Weekly Cash in Pouches Report. Value derived from Non Inventory movements across all SUs within the TP of Product **6509.** (Same as sum of **LineCCCCB** for all SUs).
Losses to/from Suspense **LineDDDD**	Branch Total	Total of individual values in suspense and stock unit columns
	Suspense	Total of all gains transferred from stock units to suspense during period. Calculated as the sum of all SU column entries for **Gains to/from Suspense** (LineCCCC)
	Stock Unit	Total nett value of losses transferred to suspense from the stock unit. Similar to **LOSSES TO/FROM SUSPENSE** total on the SU Final Balance report but excludes Cash in Pouches movements. Value derived from Non Inventory movements for the SU within the TP under Node **740** minus Product **5610, 6509** movements
Cash Rems to Suspense **LineDDDDB**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	The sum of cash transferred into pouches from the SU during the Trading Period pending dispatch from the branch. Value derived from Non Inventory movements of Product **5610.**
Transfers Out to other SUs **LineDDDDD**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Transfers of cash, cheques, stamps, foreign currency and other Value Stock to other stock units. Corresponds to **Transfers Out** total on the SU Final Balance report. Value derived from Non-inventory Figures accumulation of node **3014**
Other Payments **LineDDDDDD**	Branch Total	Total of individual values in stock unit columns

FUJ00085124

	Stock Unit	Total Value of payments made to customers and also the value of Cash Withdrawals and the use of Debit cards and Savings stamps as Methods of payment. Corresponds to **TOTAL PAYMENTS** total on the SU Final Balance report less the total of the following stock unit fields on the Branch Trading Statement: **Remittances Out Total** (LineDD) **Losses to/from Suspense** (LineDDDD) **Transfers Out to other stock units** (LineDDDDD) **Total C/Fwd** (Section E) Value derived from Non-inventory Figures accumulation of nodes **3016** (Total Payments), **964** (Clear Gain from Local Suspense) and **965** (Loss to Local Suspense), minus **740** (Losses to/from Suspense)

FUJ00085124

BTS Section E – Carried Forward Figures

LineE	Blank Line	
Cash on Hand C Fwd **LineEE**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Value of Cash held within the stock unit. Corresponds to **CASH** total on the SU Final Balance report. Value derived from Opening Figures Node **1001** accumulation
Cash Awaiting Collection C Fwd **LineEEE**	Branch Total	Same as suspense column
	Suspense	Corresponds to **C/Fwd for Cash in Pouches** total on the Suspense report. Calculated as **Sum(LineBBB+LineCCC) minus LineDDD.**
Suspense C Fwd **LineEEEE**	Branch Total	Same as suspense column
	Suspense	Cumulative C Fwd suspense amount for all items in suspense at the end of the period. Calculated as **Sum(LineBBBB+LineCCCC) minus LineDDDD**
Other MOP C Fwd **LineEEEEE**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Value of Other methods of payment held within the stock unit. Corresponds to the **MOP** less **CASH** fields on the SU Final Balance report. Value derived from Opening Figures Node **3003** accumulation less Node **1001**
ForEx CFwd **LineEEEEEE**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Value of ForEx held within the stock unit. Corresponds to **FRGN CURRENCY EQUIV** total on the SU Final Balance report. Value derived from Opening Figures Node **2016** accumulation
Other Postage C Fwd **LineEEEEEEEE**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Value of Other Postage held within the stock unit. Also includes other Stock held by Value. Corresponds to **TOTAL VALUE ITEMS & MOP** total on the SU Final Balance report less the total of the following stock unit fields on the Branch Trading Statement: **Cash C Fwd** (LineEE) **Other MOP C Fwd** (LineEEEEE) **ForEx C Fwd** (LineEEEEEE) Value derived from Opening Figures Node **3007** accumulation less Node **2016**
Total C Fwd **LineEEEEEEEEE**	Branch Total	Total of individual values in suspense and stock unit columns
	Suspense	Total of suspense column values in this section – i.e. **Sum(LineEEE, LineEEEE)**
	Stock Unit	Total of stock unit column values in this section – i.e. **Sum(LineEE, LineEEEEE, LineEEEEEE and LineEEEEEEEE)**

FUJ00085124

FUJ00085124

BTS Section F – Trading Position

LineF	**Blank Line**	
Trading Position (+/-) **LineFF**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Total of (Brought Forward Figures + Receipts) – (Payments + Carried Forward Figures) – i.e. sum of Sections B and C lines on the Branch Trading Statement minus sum of Sections D and E. This should always be zero.

BTS Section G – Local Suspense

LineG	**Blank Line**	
Discrepancy OVER Transferred **LineGG**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	A gain posted to Local Suspense when stock unit rolls over into new Trading Period. Corresponds to **Discrepancy OVER Transferred** from the SU Final Balance report. Value derived from Non Inventory Opening Figures for Node **963**
Discrepancy SHORT Transferred **LineGGG**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	A loss posted to Local Suspense when stock unit rolls over into new Trading Period. Corresponds to **Discrepancy SHORT Transferred** from the SU Final Balance report. Value derived from Non Inventory Opening Figures for Node **965**
Discrepancy OVER Resolved **LineGGGG**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	A gain cleared from Local Suspense when last stock unit rolls over into new Trading Period. Corresponds to **Discrepancy OVER Resolved** from the SU Final Balance report. Value derived from Non Inventory Opening Figures for Node **964**
Discrepancy SHORT Resolved **LineGGGGG**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	A loss cleared from Local Suspense when last stock unit rolls over into new Trading Period. Corresponds to **Discrepancy SHORT Resolved** from the SU Final Balance report. Value derived from Non Inventory Opening Figures for Node **966**

FUJ00085124

BTS Section H – Make Good and Remove Excess Cash Events

LineH	**Blank Line**	
Excess Cash Removed **LineHH**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Sum of all Remove Excess Cash 'events' during the period. Corresponds to **Excess Cash Removed** from the SU Final Balance report. Value derived from Non Inventory Opening Figures for Node **971**
Cash Shortage Made Good **LineHHH**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	Sum of all Make Good 'events' during the period. Corresponds to **Cash Shortage Made Good** from the SU Final Balance report. Value derived from Non Inventory Opening Figures for Node **972**

BTS Section I – Branch Adjustments

LineI	**Blank Line**	
Total Branch adjustments **LineII**	Branch Total	Total of individual values in stock unit columns
	Stock Unit	**Sum(LineGGGG,LineHH) minus Sum(LineGGGGG,LineHHH)**

5.1.9.3 BTS Stock Holdings

The second part of the report provides the net volume stock holding figures and a summary of the completed transaction corrections for the period. The layout is described in [BTSUI].

The section is formulated, over and above the page headings, as two Data Blocks:

- Stock Holding Data Block Headings
- Stock Holding Data Set
- Section Footer Data Block
    - Transaction Correction Summary
    - Report Declaration and Footer

The number of pages required to populate the first data block will depend upon the number of Stock Unit Volume Stock Holdings. The data block headings will be the same regardless of the number of pages required.

The production of the stock holding figures, unlike the contents of the BTS Summary Page, is driven internally by BESReports. The stock holding figures are determined from the aggregation of all volume stock holding figures across all stock units rolled to the next TP. Only non-zero holdings for each volume stock item are reported. These are found from retrieving the Opening Figures detail for each stock unit.

FUJ00085124

The contents of the Section Footer Block is made up of Transaction Correction Data and a report footer.

The production of Transaction Correction Data is achieved by extending the scope of the reference data BTSSourceData to allow line definitions reported on the stock holdings page.

Following printing of transaction correction summary a report declaration and footer are printed. The report declaration is textually driven by reference data items within BTSSourceData.

BTS Section X - Stock Volume Details

Description	Stock product description, left justified, derived from the product Medium Name
Volume	Total volume of stock held across all stock units, right justified. Calculated by adding up all the volumes of stock reported on the individual final stock unit balance reports. Each line item is derived from the sum of all like product Stock Unit Opening Figures, written at the last TP rollover.

BTS Section Y – Transaction Corrections

LineY	New page if line number > 33
LineYY	Blank line
Number of Transaction Corrections **LineYYY**	Single line showing total number of transaction corrections applied during the Trading Period. See 5.1.10.2 and [DCRHLD]. Value derived from the 'count' accumulator (RC) in the Non Inventory Opening Figures for Node **980**

BTS Section Z – Declaration and Signature

LineZ	Blank line
LineZZ	Blank line
Declaration **LineZZZ**	Text stating "I certify that the content of this balancing and trading statement is an accurate reflection of the cash and stock on hand at this branch."
LineZZZZ	Blank line
LineZZZZZ	Blank line
LineZZZZZZ	Blank line
Space for Signature **LineZZZZZZZ**	Text saying "Signature:"
End of Report marker	"*** END OF REPORT ***" (centred)

FUJ00085124

5.1.10 Adoption of Variance Handling from Discrepancies

This section defines changes impacting balancing as a result of changed requirements for the handling of variances, discrepancies and corrections defined in [DCRHLD].

Changes are required to support two specific functions, the writing of local transactions as a result of events to make good and Reclaim Cash, and Transaction Corrections.

The changes required are reflected in reference data alone.

5.1.10.1 Variance adjustment Events

As a result of an action to make good or reclaim cash following a declaration which generates a discrepancy the user may make an adjustment, recorded as an event to make good or reclaim the cash. Such an action is recorded as an event and not as an external transaction harvested to POL. Such events do however appear on the stock unit and office balance.

In order for the balance reporting mechanism to operate correctly the event must be recorded as a transaction of some description.

As a result such events will additionally be recorded as a 'Local' transaction. Such transactions will not be harvested but will contribute to the EPOSS Balancing mechanism. These transactions will have a TranType of 'L' but will have a primary mapping allowing them to be aggregated into the EPOSS Accounting Node Hierarchy and represented on the balance reports. As such Products will be generated which have primary mappings mapping into the Stock section of the accounting hierarchy.

5.1.10.2 Transaction Corrections

The number of transaction corrections that have been actioned is recorded on the BTS.

Transaction Corrections are undertaken in specific modes. In order for the number of transaction corrections to be determined, the secondary mappings associated with these modes must amalgamate into the accounting node hierarchy in order that the corrections contributing to them can be summed.

(Note that to avoid double counting, the secondary mapping is only applied to the transaction correction message, and not to its corresponding settlement message (if any). See [DCRHLD] for more details).

Thereafter the standard mechanism employed by balancing and BTS production can be used to aggregate the transactions.

FUJ00085124

5.1.11 Introduction of Extended Reprints

The implementation of four or five weekly trading periods requires changes to Report Reprints. The extended accounting period and conflict with message retention periods means that a different approach to the architecture is necessary and a different requirement for reprints to be available is adopted. Additionally new report requirements themselves means that the number of reports available for reprinting will change.

The following table denotes each report whether existing or new as part of Impact Release 3 that needs to have reprint capability attention, and denotes the attention to be given.

Report Name	Reprint Requirement
Stock Unit Balance Report	An existing report reprint capability exists which is to be continued
Cash Account Report	The Cash Account is being removed therefore the need for the reprint capability is also to be removed. However there remains a requirement to reprint the very last Cash Account Report even after the office has rolled into the first Trading Period
Branch Trading Statement	The Branch Trading Statement is a new report as part of Impact Release 3. A report reprint capability is to be introduced allowing for the reprint of the last Branch Trading Statement
Office Weekly Counter Revenue Schedule	The Counter Revenue Schedule is being removed and therefore the reprint capability is also to be removed
Office Weekly Inland Revenue Tax Credits P5589	An existing report reprint capability exists which is to be continued
Office Weekly P&A P2311MA	An existing report reprint capability exists which is to be continued
Office Weekly Redeemed Savings Stamps Summary	An existing report reprint capability exists which is to be continued
Variances Report	The Variances Report is a new report introduced as part of Impact Release 3 but subsequently withdrawn via CP3980. See [DCRHLD] for details

Two specific techniques will be adopted to ensure the reprint capabilities required for Impact Release 3 are implemented successfully and any migration implications are resolved.

- Change weekly report reprint selection to be implemented by Date Range as opposed to CAP for any report that is not aligned to the accounting period but rather to a week. Namely the Weekly reports above, and also including the Variances Report.
- Implement a BLOB technology based solution for any report reprint where the report is aligned to a specific accounting period. Namely the Stock Unit Balance and Branch Trading Statement.

The changes involved in the above techniques and required to implement report reprints as part of Impact Release 3 are now described.

FUJ00085124

5.1.11.1 Report Reprints by Date Range

All Report Reprints that require a weekly report to be reprinted will adopt the approach of selecting the report instance to be reprinted by date range. The report reprints impacted are therefore:

- Counters Revenue Schedule
- Inland Revenue Tax Credits P5589
- P&A P2311MA
- Redeemed Savings Stamps Summary

Changes are to be made to the selection criteria for the above reports so that instead of presenting a CAP selection button, a Date Range selection is presented. The available scope for reprinting weekly reports shall be limited to the 5 previously produced reports, ie. 5 weeks.

The Date Range supplied will be used to determine the report to be reprinted and the existing mechanism of markers used to reproduce the report. Note that the start/end dates presented on the selection picklist are based on when the original reports were produced and cutoff (which may or may not be weekly).

5.1.11.2 BLOB Reprints

The Stock Unit Balance Report and the new Branch Trading Statement will adopt a different approach to reprinting from now on. Rather than retrieving the raw data and regenerating the report with identifying reprint information the original report will be saved/stored in its originally printed form and reprinted from there.

At the time the source report is produced the software will produce a 'copy' of the report in a persistent object (BLOB) in the collection '**Reprints**'. The object will be keyed on the report. Riposte will automatically perform data compression of the target report if the report size is greater than 1024 bytes.

The stated requirement is merely to produce the last produced stock unit balance and Branch Trading Statement, hence there is only a need for one object instance for any one report. However the changes required to the existing functionality simply allow for reprints to be provided for all Balance Periods rolled over in the last accounting period. Hence stock balance reprints will be available for the last accounting period, all balance periods in that period and all balance periods in the current accounting period.

For branch trading statement reports, the reprint object is called '**BTSCAP<nn>**'. For stock unit balance reports, the reprint object is called '**SUBalance**<*su*>**-CAP**<*nn*>**-BP**<*m*>' where <*su*> is the stock unit name, <*nn*> is the CAP/TP number, and <*m*> is the BP number.

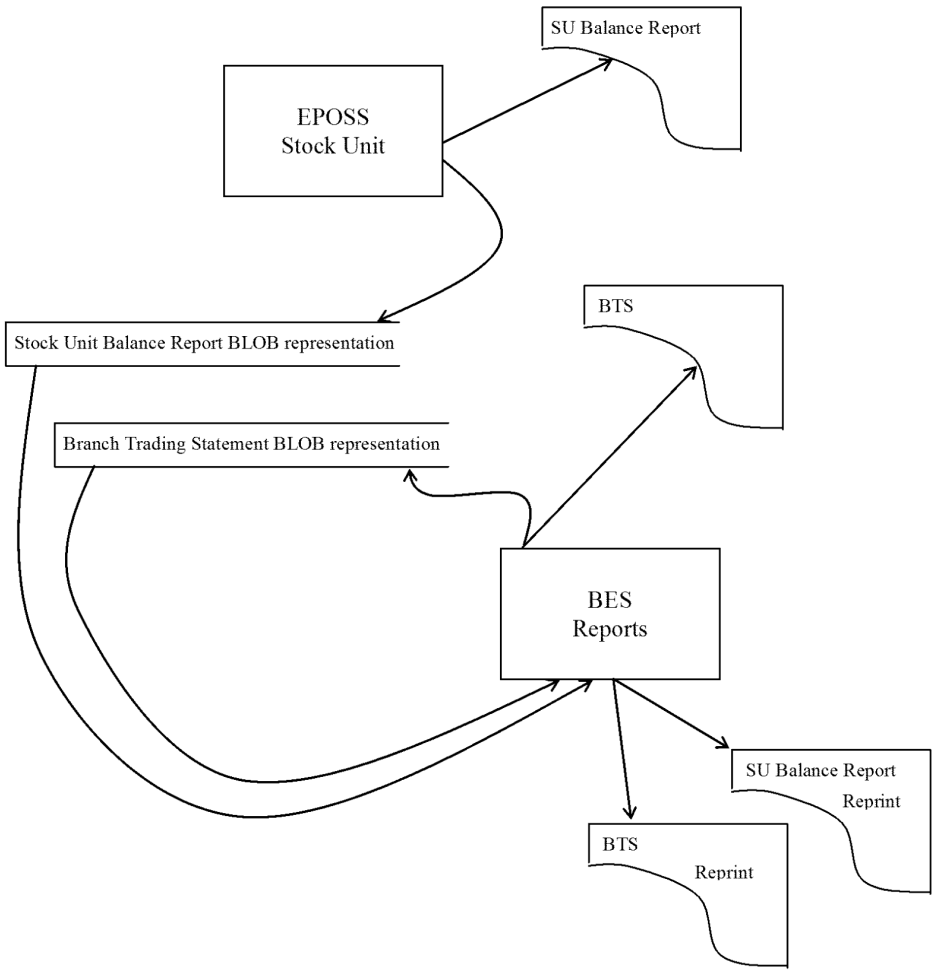
In principle old reprint objects can be pruned, since they are not required beyond the following TP. However, since retaining old reprint objects may be useful for diagnostic purposes, and the amount of space consumed is not large per report and the names are recycled each year, no pruning is currently proposed.

Upon selecting a reprint of the report the persistent object will form the source to the report reprint. The EPOSS component BESReports will entirely handle the print of report reprints whilst it is the source components' of for the reports that handle generation of the reprint data.

FUJ00085124

Whilst the BLOB contains the report in a form that can easily be reprinted, without substantial knowledge of the original report format, two considerations have to be given. Firstly whilst Riposte performs compression to data stored in BLOBs there may be a need to provide further compression. Secondly reprinted reports for the most part also indicate the reprint by offering a 'Reprint' banner and date and times adjusted to the point at which the reprint is printed. As a result the Blob mechanism will offer parameter elements to enable the reprint to be so annotated.

The following diagram portrays the solution.



5.1.11.3 Reprinting the Last CAP SU Balance and Cash Account

In addition to making available the appropriate changes to reprint the Stock Unit Balance and Branch Trading Statement the service must also take account of the ability to be able to produce the last Stock Unit Balance Report in the old format, once the Stock Unit has rolled into the first TP, and to be able to re-produce the last Cash Account, once the office has rolled into the first TP.

Tied in with this requirement is the need to ensure the Button Icons represent what will be reprinted. The following table defines the potential scope.

SU Status	Office Status	Report Available for Reprint	Button Icon

FUJ00085124

CAP	CAP	Existing functionality	Existing functionality
First TP	Last CAP	SU Balance of last CAP + BPs in current TP	SU Balance
		Existing functionality	CA Reprint
	First TP	SU Balance of last CAP + BPs in current TP	SU Balance
		Cash Account of last CAP only	CA Reprint
Subsequent TP	First TP	SU Balance of previous TP + BPs in current TP	SU Balance
		Cash Account of last CAP only	CA Reprint
	Subsequent TP	SU Balance of previous TP + BPs in current TP	SU Balance
		BTS of previous TP only	BTS Reprint

The migration requirements must take account of these circumstances.

5.1.12 Changes to Suspense Account

The operation of the suspense Account is changing as a result of the introduction of Impact. Release 3.

The concept of a Local Suspense Account is introduced. Further the so called ## figures for the office which enable the office suspense position to be determined, can no longer be maintained, as determination of suspense account items is based upon products that have a cash account mapping to Tables 2, 2a and 3 in the cash account.

Additionally suspense items contribute to the new Branch Trading Statement and hence mechanisms will be required to support the output to the BTS.

Two sets of changes are required to the operation of Suspense:

- A new set of ## figures will be produced to support the rollover of the office into which the aggregated stock unit suspense movements will be maintained. The ## figures can then be used to support production of the BTS and Suspense Account Reports.
- The concept of a local suspense account will be introduced. Discrepancies outstanding at each Stock Unit Rollover will be moved to the local suspense account, and the net amount in local suspense when the last stock unit is rolled remains, prohibiting its rollover, until it is acted on.

5.1.12.1 Introduction of new Hash Hash Figures

The main function of the existing ## figures is in support of cash account production. Additionally they are used to provide the brought forward figures to the suspense account report. In this capacity the identification of suspense products is made by reference to their cash account mappings. However the removal of the cash account means that suspense products can no longer be identified from their cash account mappings. A new mechanism is

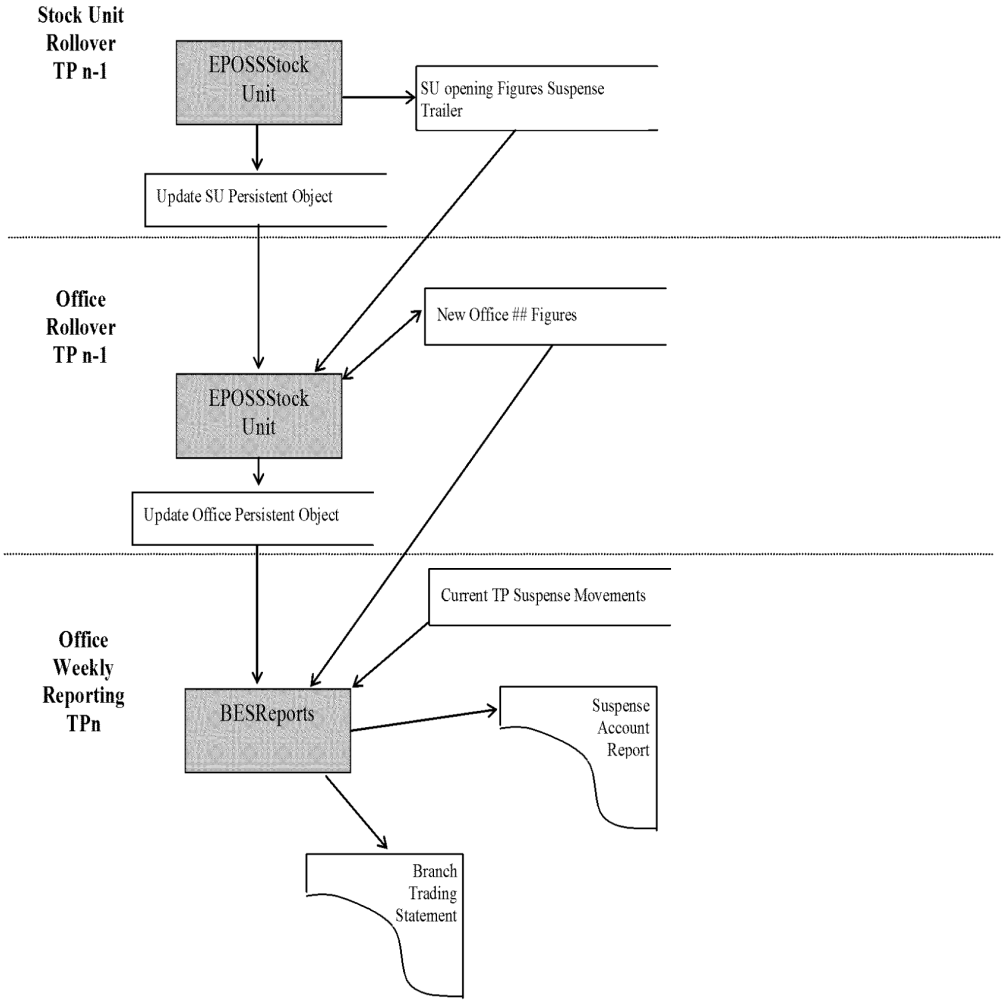
FUJ00085124

required. Additionally the suspense figures, both brought forward, and movements within a trading period contribute to the BTS. A mechanism is required to support this.

The concepts of Stock Unit suspense trailer and office (##) suspense figures will be introduced at stock unit and office rollovers respectively.

The stock unit suspense trailer will record the movements of suspense items within the Trading Period. At office rollover the movements for all stock units will be aggregated into the ## suspense figures. Suspense Movements are identified by a Primary Mapping node of either 490 or 740.

The following scheme will apply.



A new set of Stock Unit Rollover Figures will be created, grouped as the Suspense Trailer. The figures record suspense movements for the stock unit for the trading period. The Office Rollover takes each set of these figures and amalgamates them into new ## records at the Office Rollover for the TP, bringing forward the office ## records from the previous period.

The ## figures provide the Suspense Account Report in the next TP with its opening movements. Production of the Suspense Account Report is completed by taking the suspense

FUJ00085124

account movements for the current TP also into account. There is no change to the latter processing.

Additionally suspense movements must be available to support the BTS. Figures are required in support of the BTS. The items required for the BTS can be supported from the Suspense Trailer records and the Office ## Figures.

The Suspense Brought Forward Figures are provided for the BTS in TPn from the Office ## figures in TPn-1.

The Suspense movement figures for the BTS in TPn are provided as an aggregate of all the suspense trailer figures from the SU Rollover for TPn.

The Suspense Movement figures for the BTS in TPn are a simple sum of the brought forward figures for suspense, and the TP movements of suspense, within the same report.

In order to implement suspense on the BTS two new attributes to the BTS Source Data parameters are needed. The first is an Office attribute. It denotes that the line contains an office only figure and hence is derived from a message held against the office rollover persistent object. This is explained in more detail in section 5.1.9.

5.1.12.2 Introduce Local Suspense

A local suspense account is introduced as part of Impact Release 3. As part of the stock unit rollover process into a new TP any amount of discrepancy will be moved to local suspense. On the last stock unit rollover if there is any net discrepancy left then the user will be required to clear it before the rollover can be completed..

The movement of discrepancies to local suspense will be made as a contra transaction to the discrepancy in the stock unit and a balancing transaction in housekeeping mode to a local suspense product.

On the final stock unit all transactions posted with either local suspense products will be retrieved for all stock unit rollovers into the next TP and the sum local suspense total determined.

If the sum total is zero then the final stock unit can continue to rollover. If however the total is non-zero, then local suspense must be cleared before rollover can continue.

5.1.12.2.1 Clearing Local Suspense

Local Suspense can be cleared through transactions undertaken on the housekeeping menu or through the dialogue presented at the point of stock unit rollover.

The mechanism is described in EA/IFS/013.

5.1.12.3 Suspense Account Reports

The existing single Suspense Account Report is to change. Firstly the report group detail for Cash In Pouches is to be summarised and the detail lines moved to a new suspense report that handles Cash In Pouches alone. Secondly a new group is to be added to the Suspense Account Report to report on Local Suspense Movements within the Trading Period. Thirdly, any group with zero brought forward figures or movements will be suppressed.

As part of Impact, POL are withdrawing the use of a number of suspense products, but existing positions for these are valid, so remain in the report until cleared. However, there are

FUJ00085124

a couple of new suspense product pairs (Emergency Txn Payments and Receipts), and these will be added to the Suspense report as a new section at the end.

The changes to split the suspense report are directly at the instruction of CP 3787. The report layouts are provided in [REPREC].

The above changes will all be introduced at Point 20 in the migration path (although the new suspense products will not be transacted until point 50).

All changes affect the counter component BESReports.

The current structure of the suspense report is provided by the reference data collection <Collection:SuspenseGroups>, which corresponds with the cash account mapping structure of each suspense item. Each group on the suspense report corresponds to a line on the cash account for suspense items, and an object in the Suspense Groups collection. The order of the objects in the collection reflects the cash account mappings and the order required on the suspense report.

The change of order and introduction of new suspense groups that will not have corresponding cash account mappings requires a new structure to support the suspense account.

A new collection <Collection:SuspenseSections> is introduced, see Section 12.9, that will support the structure of both the Suspense Report and the new Cash In Pouches Awaiting Collection Report. Rather than providing the definition of each group by its corresponding cash account mapping, the collection defines an object by an arbitrary value that put together represent the order of the reports' groups or sections. The collection replaces the use of the collection <Collection:SuspenseGroups>, which will no longer be used after point 20.

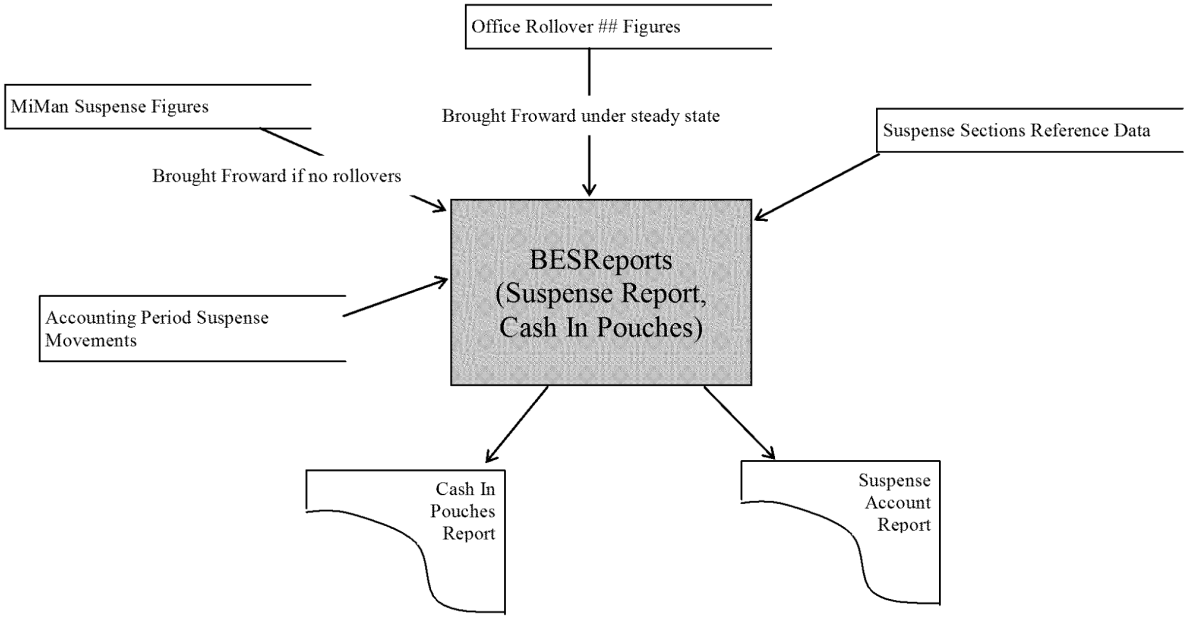
The collection provides the vehicle to identify each section on the reports and provide the mapping to the source transactions that provide the data for each section. The objects within the collection have arbitrary identifiers but together the sequence of the objects provides the sequence of the sections (groups) on the reports. Each object identifies the section heading together with, by product identification, the products for which transactions present will populate each group. A group will only be printed if there are transactions posted.

Both reports provide for each section the net brought forward movement value, the period movements individually or summarised, and the net carried forward value. The removal of the cash account as part of Impact means that carried brought forward figures can no longer be sourced from the previous period's cash account data held in the message store. However the possibility of suspense data being provided from a migrated office immediately in advance of migration point 20 means that the reports must take account of migrated data, which actually is stored in a format akin to cash account data.

The reports will only consider brought forward suspense data from migration when no following accounting period rollover has taken place. In steady state brought forward figures will be determined from the last rollover's ## figures.

The following diagram depicts this:

FUJ00085124



5.1.12.4 Cash In Pouches

The introduction of the Branch Trading Statement as part of Impact Release 3 requires movements of Cash In Pouches to be reported. This presents some problems with the current implementation of Cash In Pouches. The problems are caused by the fact that there is only one product, 5610, that represents movements, both of cash into pouches and awaiting collection, and the collection of pouches from the branch. However the BTS is required to report movement separately depending whether it is movement into the pouch or collection from the branch.

As a result the implementation of cash in pouches is to change so that a separate product is used depending upon the movement.

The change will come into effect at Migration Point 20 of Impact Release 3 but the dependency upon reference data will mean that data changes will be made ahead of that point.

A new product 6509 will be introduced in reference data that reflects the movement of cash in pouches, being collected from the branch. The existing product 5610 will continue to be used for movements of cash into pouches and awaiting collection. The new product will be assigned the cash account mappings of product 5610, hence the movements as currently reported to the cash account will be as at present. The new product will have the reverse accounting sense to product 5610. The new product will be introduced before migration point 20 but will be benign.

A change is required to LFS to change the Remittance-Out (Despatch Pouches) code to then enforce use of this new product rather than product 5610. Essentially when an RODP transaction is performed LFS must determine the RODP product number to use rather than transacting product 5610 with a negative sign of the source ROSP transaction. The new product will already have the reverse sense to product 5610 and hence there is no need to reverse the sign. This is achieved by accessing <Collection:CounterConfigParams>, <ObjectName:Counter> attribute <CashInPouchesProduct:>, which identifies the product number.

FUJ00085124

The changes to the code will be implemented as part of Point 20 migration, hence all reference data changes before that point will be benign. However, additionally, if the code fails to find the <CashInPouchesProduct:> attribute, or the contra product identified by the attribute, then the code will act is without any change to Cash In Pouches processing.

In the event of a ROSP transaction being reversed by a RISP transaction then product 5610 will continue to be used.

The following table summaries the transaction movement scheme.

Transaction	Mode	Product Before Change	Product After Change
Cash Awaiting Collection	ROSP	5610	5610
Cash Awaiting Collection, reversed	RISP	5610	5610
Despatch Pouches	RODP	5610	6509

FUJ00085124

5.1.13 Adoption of Cut Off Reports for Trading Periods

As a result of the changes replacing the Cash Account Period by a Trading Period a number of reports will be changed to become date range based rather than accounting period based. This tends to apply to weekly reports where the use of the CAP was a fortunate vehicle to which to align the report. Extending the accounting period to 4 or 5 weeks requires that these reports are still produced weekly. However, they will still be cut off by a Trading Period rollover.

5.1.14 Amendment/Removal of Generic Reports

Changes to reports and receipts over and above the specific report changes to the Suspense Account, introduction of the BTS, changes to the balance reports and changes to Giro Reports are documented in EA/IFS/011.

No specification of the further report changes is provided here.

5.1.1.1 Giro Reports

A specific change to the content layout of Giro Reports will be implemented as part of Impact Release 3, as a result of CP 3842 and CP 3888. A change is required to Giro Reports, the production of reports their alignment to a CAP, and necessarily provides a week number relationship. With the removal of accounting on a weekly basis there is still a need to produce giro reports aligned to some week number rather than being aligned to a Trading Period.

As a result the printing of the Cash Account Week Number on Giro Reports will replaced by an arbitrary defined week number.

The change affects those giro reports that are defined as client reports and are produced daily. The specific reports impacted are therefore:

- G9901MA Daily Record of Giro Deposits
- G9902MA Daily Record of Giro Withdrawals
- P5589 Inland Revenue Tax Credits (and reprint option)
- Counter Weekly Inland Revenue Tax Credits
- P2311MA P&A Summary (and reprint option)
- P2311MA (B) Batch Control Voucher
- Counter Weekly Green/Violet Giros
- Counter Weekly P&A
- Office Weekly P&A

Instead of printing the Cash Account Period Number these reports will now print a calculated week number. The week number will be assigned based upon the date the report is produced, aligned to a calendar starting at the beginning of the POL financial year. Prior to Point 50, this is in effect the current CAP week

FUJ00085124

5.2 Interface

The user interfaces are defined within references [RPUI], [DCRUI] and [BTSUI]. An Appendix will be provided that defines the format of the messages relayed to the agent and other systems where changed. [RDMCHLD] and Appendix B – Affected Reference Data Collections define the reference data that drives the application.

5.3 Distributed Application Services

There are no changes to Distributed Application Services as a result of the requirements impacting this design.

5.4 Information Management

Standard facilities will be used.

5.5 Networking Services

These will be provided by standard Riposte messaging.

5.6 Platforms

This document only addresses the counter platform.

5.6.1 Live Counters

Impact Release 3 will be supported on live counters.

5.6.2 Training Mode

Impact Release 3 will not be supported in training mode, a change resulting from CP 3842. Access to Training Mode will be removed, by removal of the desktop button.

5.6.3 Training Counters

Impact Release 3 will be supported on training counters.

FUJ00085124

6.0 Systems Management

The system is primarily controlled through reference data.

6.1 Reference Data

The implementation of Impact Release 3 uses reference data to replace the cash account calendar by one for Trading Periods. The concept of Tertiary Mappings is introduced, implemented as Product Reference Data. Configuration parameters are also added as reference data to control Report Legends and Expiry Periods.

6.2 Receipts and Reports

These are controlled through the normal Global Objects mechanism.

6.3 Screens and Dialogue Flows

For screens and dialogues, see references [RPUI], [DCRUI] and [BTSUI].

The UI is implemented according to the standards defined in [STYLE].

The Menu structure supporting users' access to functions is provided in [MENU], [MENU2].

FUJ00085124

7.0 Application Development

The development will follow a joint working exercise with design to incrementally build the changes concurrently with design progress using a controlled RAD approach.

Each significant development will have a low-level design.

During development, source will be held in VSS. VB6 or later versions will be used as appropriate. PVCS will be used for delivery, as normal.

All source code will be documented.

FUJ00085124

8.0 System Qualities

The primary aim of the solution adopted for Impact Release 3 Balancing and handling stock by volume is to ensure the risks associated applying the required changes are mitigated as far as possible. Additionally the following System Qualities are considered.

8.1 Availability

The counter system is available during the prescribed post office opening hours. No changes to system availability are perceived as a result of Impact Release 3.

8.2 Usability

The system has been designed for use by users already familiar with the Horizon system. Changes to the Balancing process as a result of the introduction of Trading periods necessitate changes to branch processes. However a more significant impact on processes is attributable to the introduction of handling stock by volume rather than value. There will be significant process changes, however the look and feel will remain the same.

8.3 Supportability

No changes to system supportability are perceived as a result of Impact Release 3.

However additional events will be reported to the event log, as per coding standards, to enhance available diagnostic information.

8.4 Security

No changes to system security are perceived as a result of Impact Release 3.

8.5 Potential for Change

The changes required for Impact Release 3 are invasive to the existing EPOSS product. The potential for change is therefore limited. However by applying an incremental approach to the design any future change should be more easily manageable.

FUJ00085124

9.0 Solution Implementation Strategy

9.1 Migration

Migration of PO outlets to the new handling of Stock by Volume and Branch Trading Periods is documented through the Migration High Level Design in [MIGHLD]. This document provides a more detailed view of the Migration Design elements that apply to the changes for Balancing, Rollover and Stock Handling by Volume.

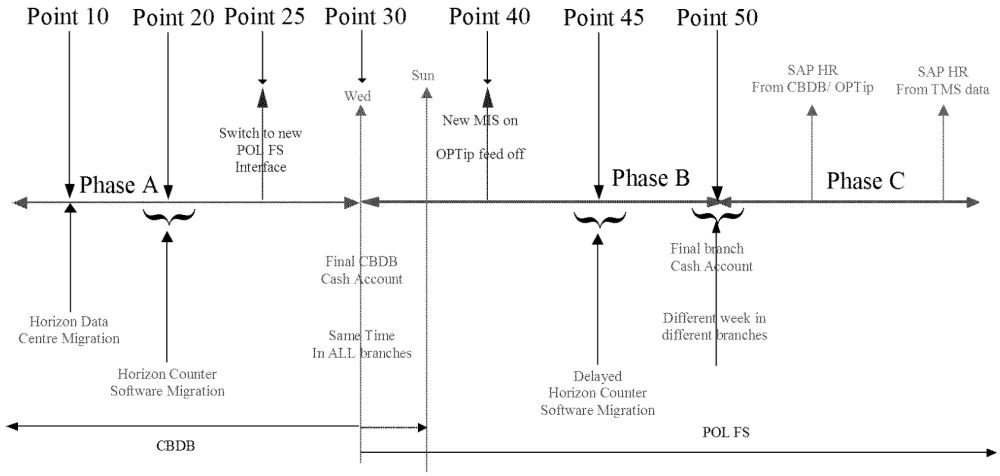


Figure 12 - Migration Timeline

First of all, S80 counter code, and reference data to support it must be delivered to the counters. These may occur in parallel over a period prior to point 30 and must be independent – i.e. S80 code must emulate pre-S80 functionality whether or not the reference data has arrived, and pre-S80 code must not be affected by the S80 reference data.

The migration impacts of each function change are documented in the following sections.

9.1.1 Extended Retention Periods

Changes associated with the Extending Retention Periods must be activated before an outlet rolls a Stock Unit into the first Trading Period. Failure to do so will allow a stock unit to perform transactions in the first trading period which could expire before they are needed to be taken into account in a rollover to the next accounting period.

These changes are to be implemented as soon as the S80 Software is installed (migration Point 20) and no Soft Launch control will be associated with this functionality.

9.1.2 Introduction of Trading Period Rollovers

Migration from operating in CAP mode to operating in TP mode requires a number of changes to be made at Points 20 and 50 in the migration timeline (see Figure 12).

FUJ00085124

(Note that various other counter changes need to be made at Point 30, but those are not in the scope of this particular section).

The 'Point 50' changes will be initiated by rollover during the day from a designated branch-specific 'Final CAP' to a calculated 'First TP', so do not simply happen overnight as a result of new reference data. Nevertheless, reference data has a major role to play, since it defines

- the particular 'Final CAP' value for the branch
- the replacement TP-mode menus and legends to be used once the current Stock Unit, or the Office as a whole, has rolled into the 'First TP'.
- the original CAP-mode menus and legends to be reinstated if the user attaches to a different Stock Unit that has not yet rolled into the 'First TP'

The designated 'final CAP' values are made available via SoftLaunch definitions (see [SOFTLAUNCH]) that depend on corresponding branch-specific trigger products.

For more details, see [MIGHLD]

FUJ00085124

10.0 Costs, Risks and Timescales

Costs are dealt with in the plans.

Timescales are documented in the plans.

Consideration to the design from the outset has identified two areas of attention requiring analysis for risk:

- Technical Risks
- Management Risks

Every effort has been made in the production of this high level design to mitigate against the unnecessary overheads that could be caused by these risks inconsideration of the design.

10.1 Technical Risks

Two major Technical Risks have been identified in consideration to the design.

Firstly the migration implications of the design proposed in [DP] and defined by the high level design here are significant. For example it has already been identified that a new process for the migration of new outlets onto the Branch Trading Period regime will be required.

Secondly the changes to the counter applications, and in particular EPOSS, are invasive. The EPOSS Application does not lend itself easily to invasive change. Its component elements are few in number but a single element may have multiple uses in the functioning counter, hence a change can have an impact in areas that are not required for change in a business sense. There is hence a very real risk of regression in any change made to EPOSS under the changes required for Impact Release 3.

No other technical risks have been identified so far.

10.2 Management Risks

The following programme management risks have been identified in consideration of the design.

1. The timescales for delivery of S80 are recognised to be tight and if at all possible it is required that development should commence before the design (which in itself is substantive) is brought to final approval.
2. The high level design is one of two complementary HLDs for changes to the Counter in response to Impact Release 3. Ideally the two documents hence should be levelled to the same degree and as stated, complement each other. This may not be as easy as perceived given the different resources and timescales for delivery.
3. The high level design approach to deliver design definitions which may be developed independently of each other can lead to potential nugatory development if the changes are not considered. For example the change to follow the use of Trading Periods, whilst retaining the cash account involves a degree of nugatory work as later the cash account would be removed anyway. It is considered a subject of planning as to how the nugatory work is kept to a minimum.
4. The layout of the new Balance Report and Branch Trading Statement may need revision as a result of customer feedback, potentially involving significant rework.

FUJ00085124

11.0 Appendix A – Design Proposal Cross Reference Matrix

The following compliance matrix resolves each reference provided in [DP] to ensure all Counter requirements are covered by each of the Counter HLDs and to resolve the functional unit responsible for providing the balancing functionality.

Design Proposal Reference	Sub-Reference	Responsible HLD	Functional Unit	Notes
2.5.1.1				
2.5.1.1.1	2.5.1.1.1.1	This Document	Extension of Transaction Retention (5.1.1)	
	2.5.1.1.1.2	This Document	Extension of Transaction Retention (5.1.1)	
	2.5.1.1.1.3	This Document	Extension of Transaction Retention (5.1.1)	
	2.5.1.1.1.4	[DCRHLD] and this document	Protection against Loss of Data(5.1.1.4)	
2.5.1.1.2	2.5.1.1.2.1	This Document	Amendment of Transaction Attributes for Volume Stock (5.1.4)	
	2.5.1.1.2.2	This Document	Amendment of Transaction Attributes for Volume Stock (5.1.4)	
	2.5.1.1.2.3	[DCRHLD]		
	2.5.1.1.2.4	This Document	Amendment of Transaction Attributes for Volume Stock (5.1.4)	
2.5.1.1.3	2.5.1.1.3.1	This Document	Amendment of Transaction Attributes for Volume Stock (5.1.4)	
	2.5.1.1.3.2	This Document	Amendment of Transaction Attributes for Volume Stock (5.1.4)	
	2.5.1.1.3.3	This Document	Amendment of Transaction Attributes for Volume Stock (5.1.4)	
	2.5.1.1.3.4	This Document	Amendment of Transaction Attributes for Volume Stock (5.1.4)	
	2.5.1.1.3.5	This	Amendment of	

FUJ00085124

		Document	Transaction Attributes for Volume Stock (5.1.4)	
2.5.1.1.4	2.5.1.1.4.1	This Document	Changes to Suspense Account (5.1.12)	
2.5.1.1.5		This Document	Amendment of Transaction Attributes for Volume Stock (5.1.4)	
2.5.1.1.6		This Document	Amendment of Transaction Attributes for Volume Stock (5.1.4)	
2.5.1.1.7		[DCRHLD]		
2.5.1.2				
2.5.1.2.1	2.5.1.2.1.1	[DCRHLD]		
	2.5.1.2.1.2	[DCRHLD]		
	2.5.1.2.1.3	[DCRHLD]		
2.5.1.2.2	2.5.1.2.2.1	[DCRHLD]		
2.5.1.2.3		[DCRHLD]		
2.5.1.2.4		[DCRHLD]		
2.5.1.2.5		[DCRHLD]		
2.5.1.3				
2.5.1.3.1		[DCRHLD]		
2.5.1.3.2		This Document	Introduction of Stock Balance Reports (5.1.8)	
2.5.1.3.3		This Document	Changes to Suspense Account (5.1.12)	
2.5.1.3.4		This Document	Adoption of Cut Off Reports for Trading Periods (5.1.13)	
2.5.1.3.5		[DCRHLD]		
2.5.1.3.6		This Document	Amendment/Removal of Generic Reports (5.1.14)	
2.5.1.3.7	2.5.1.3.7.1	This Document	Amendment/Removal of Generic Reports (5.1.14)	Except Balance Reports undertaken with 2.5.1.3.2
	2.5.1.3.7.2	This Document	Amendment/Removal of Generic Reports (5.1.14)	
	2.5.1.3.7.3	This Document	Amendment/Removal of Generic Reports (5.1.14)	
2.5.1.3.8		This		

FUJ00085124

		Document	Introduction of Extended Reprints (5.1.11)	
2.5.1.3.9		This Document	Amendment/Removal of Generic Reports (5.1.14)	
2.5.1.3.10		This Document	Amendment/Removal of Generic Reports (5.1.14)	
2.5.1.3.11		This Document	Giro Reports (5.1.14.1)	
2.5.1.4			Introduction of Trading Period Rollovers (5.1.2) Introduction of Aggregation Engine for Volume Stock (5.1.3) Introduction of Volume Stock Rollover Data Model (5.1.5)	
2.5.1.4.1	2.5.1.4.1.1	This Document	Introduction of Trading Period Rollover Script (5.1.6)	
	2.5.1.4.1.2	This Document	Adoption of Variance Handling from Discrepancies (5.1.10)	
	2.5.1.4.1.3	This Document	Introduction of Stock Balance Reports (5.1.8)	
	2.5.1.4.1.4	This Document	Introduction of Volume Stock Balancing Model (5.1.7)	
2.5.1.4.2		This Document	Introduction of Branch Trading Statement (5.1.9)	
2.5.1.4.3		This Document	Introduction of Branch Trading Statement (5.1.9)	
2.5.1.4.4		This Document	Management of Steady-State TP Rollover **(5.1.2.5)**	
2.5.1.5				
2.5.1.5.1		[DCRHLD]		
2.5.1.5.2		[DCRHLD]		
2.5.1.5.3		[DCRHLD]		
2.5.1.5.4		[DCRHLD]		
2.5.1.5.5		[DCRHLD]		
2.5.1.5.6		[DCRHLD]		

FUJ00085124

2.5.1.6				
2.5.1.6.1		[DCRHLD]		
2.5.1.6.2		[DCRHLD]		
2.5.1.6.3		[DCRHLD]		
2.5.1.6.4		[DCRHLD]		
2.5.1.7				
2.5.1.7.1		[DCRHLD]		
2.5.1.7.2		[DCRHLD]		
2.5.1.7.3		[DCRHLD]		
2.5.1.8		This Document	MiMAN Migration **(5.1.2.8)**	
2.5.1.9		This Document	Training Mode **(5.6.2)**	

FUJ00085124

12.0 Appendix B – Affected Reference Data Collections

This appendix provides a reference to the affected reference data collections as a result of changes to Balancing, Rollover and Handling Stock by Volume at S80. Each collection is address by a separate sub section.

12.1 AccountingPeriods Collection

This is a new collection, defined in [RDMCHLD], and an example is reproduced here for convenience

```
<Collection:_AccountingPeriods>
<ObjectName:2004_00>
<StartDate:01-Jan-2004 00:00:01>
<EndDate:>
<RData:
        <Data:
           <FinYear:
                  <SD:23/03/2004 12:12:34>
                  <ED:21/03/2005 23:59:59>
                 <AP:
                  <1:
                         <SD:23/03/2004 12:12:34>
                         <ED:20/04/2004 23:59:59>
                  >
                  <2:
                         <SD:21/04/2004 12:12:34>
                         <ED:18/05/2004 23:59:59>
                  >
           Etc to include all APs for the Financial Year
           >
        >
>
```

12.2 OutletDetails

The OutletDetails object within the 'Outlet' collection Outlet is amended to include a new attribute 'BTSOffset' that indicates the branch trading statement offset. The syntax is defined in [RDMCHLD], and an example reproduced here for convenience

```
<Collection:_Outlet>
<ObjectName:OutletDetails_03>
<StartDate:25-JAN-2001 10:01:46>
<EndDate:>
<RData:
        <Data:
                <FadCode:1764209>
                <Add1:6 High Street>
                <Add2:Bromborough>
                <Add3:>
                <Add4:Wirral>
                <Add5:Merseyside>
                <Postcode:CH62 7HA>
                <CATypeCode:2>
```

FUJ00085124

<LanguageCode:1>
<ContractTypeCode:SPSOCA>
<Name:Bromborough>
<CATel: GRO >
<PWTel:>
<PWTelOutOfHours:>
<RegionCode:853>
<PlantCode:R782>
<BTSOffset:0>
>
>

12.3 CounterConfigParams Collection

A new collection of CounterConfigParams is introduced as TypeC Reference Data. It will define counter application configuration parameters, introduced at specific releases.

Example:

<Collection:CounterConfigParams>
<ObjectName:Counter>
<StartDate:01-JAN-2004 00:00:00>
<EndDate:>
<RData:
<Data:
<TxnExpiry:42>
<MinAppExpiry:20>
<RollExpiry:38>
<CAPLegend:CAP>
<TPLegend:TP>
<CashInPouchesProduct:6509>
>
>

The following attributes are currently specified.

Field Name	Description	Format	Comments
Collection		String	CounterConfigParams
ObjectName	Default Object for Counter, but specific objects as required by individual counter applications if and when required; eg APS. The Counter object provides system (counter application) wide variable values which may only be overwritten by Application specific values	String	Counter
StartDate	Effective Start Date		
EndDate	Effective End Date		
RData			
Data			
TxnExpiry	Message Expiry Value	Numeric	Currently assigned a value of 42 See Extension of Transaction Retention (5.1.1)

FUJ00085124

MinAppExpiry	Message Expiry Value	Numeric	Currently assigned a value of 20 See Extension of Transaction Retention (5.1.1)
RollExpiry	Message Expiry Value	Numeric	Currently assigned a value of 38. See Extension of Transaction Retention (5.1.1)
CAPLegend	Report and Dialogue Accounting Period Legend to use prior to the introduction of Branch Trading Periods	String	CAP
TPLegend	Report and Dialogue Accounting Period Legend to use after to the introduction of Branch Trading Periods	String	TP
CashInPouchesProduct	Identifies the Identifier that denotes the product nominated to be assigned in transactions for the removal of cash pouches from the branch	Numeric	

12.4 EPOSSNodes Collection

The existing collection EPOSSNodes is impacted. New Node instances will be defined to support the Tertiary Mapping structure for reporting the sale of stock managed by volume. New accumulators will require to be added to the Primary Mapping node structures affected by those same products so that volume stock quantities can be separately managed from those of value stock.

Example:

```
<Collection:_EPOSSNodes>
<ObjectName:10_07>
<StartDate:01-JAN-1996 00:00:31>
<EndDate:>
<RData:
        <Data:
                <NID:10>
                <NN:OTHER BANKS CHEQUES>
                <L:2>
                <C:13>
                <C:15>
                <A:
                        <N:10>
                        <AN:LTSV>
                        <Att:LTSV>
                        <Pos:True>
                        <F:Sum>
                        <Con:0>
                >
                <A:
                        <N:10>
```

FUJ00085124


```
                <AN:Qty>
                <Att:Qty>
                <Pos:True>
                <F:Sum>
                <Con:0>
        >
        <A:
                <N:10>
                <AN:RC>
                <Att:RC>
                <Pos:True>
                <F:Count>
                <Con:0>
        >
        <A:
                <N:10>
                <AN:SV>
                <Att:SV>
                <Pos:True>
                <F:Sum>
                <Con:0>
        >
>
```

The following attributes are currently specified.

Field Name	Description	Format	Comments
Collection		String	EPOSSNodes
ObjectName	Numeric Identifier identifying object node	Numeric	
StartDate	Effective Start Date		
EndDate	Effective End Date		
RData			
Data			
NID	Node ID: Node Identifier. As well as being the object name	Numeric	
NN	Descriptive name for this Node. This is the text that is displayed on a report. It should not be > 30 characters otherwise it may be truncated at the end	String	
L	Node Level: Level in the hierarchy at which this Node sits. (1; 2; 3; 4; 5)	Numeric	
C	Optional Repeating Attribute, Child Node number: If this node has any children, these are listed using repeated occurrences of this attribute. If this node has no children then the attribute is not used. The Child Nodes must be present in ascending numeric sequence	Numeric	
A	Repeating group describing all accumulators associated with this node.		
N	The Node number is listed again when	Numeric	

FUJ00085124

	defining accumulates		
AN	Accumulator Name: Defines what is to be accumulated; e.g. Record Count (RC), Sale Value (SV), Quantity (Qty)	String	SV, QTY, RC, SVETM, QTYETM, RCETM, SVNTM, QTYNTM, RCNTM, LTSV
Att	Attribute: Attribute this accumulator works on. Defines the attribute that is searched for within MessageStore (usually RC; SV; Qty; LTSV; SI)	String	
Pos	Positive Effect: Indicates if this accumulator totals in a positive or negative way. Used when a negative number (e.g. refund) is to be displayed as a positive number; if false is specified, the absolute value will be used	String	True or False
F	Function: Indicates how totals are derived using this accumulator. Defines whether the accumulated value should be summed or counted; e.g. the sale value (SV) should be summed to give total value of all transactions, the record count (RC) should be counted to give total number of transactions	String	Sum, Count, SumETM, SumNTM, CountETM, CountNTM
Con	ContextID	Numeric	Usage Unknown

12.5 EPOSSDNodes Collection

The existing collection EPOSSDNodes is impacted. New accumulators will require to be added to the dynamic node specifications used by those Aggregations and Reports which require to accumulate Volume Stock separately.

Example:

```
<Collection:_EPOSSDNodes>
<ObjectName:5001_02>
<StartDate:01-JAN-1996 00:00:11>
<EndDate:>
<RData:
        <Data:
                <DN:Price Grouping>
                <GB:EPOSSTransaction.SaleValue>
                <GI:Every>
                <GSS:Forward>
                <GST:Number>
                <GID:5001>
                <C:>
                <A:
                        <N:5001>
                        <AN:Qty>
                        <Att:EPOSSTransaction.Qty>
                        <Pos:True>
                        <F:Sum>
```

FUJ00085124


```
                <Con:0>
            >
            <A:
                <N:5001>
                <AN:RC>
                <Att:EPOSSTransaction.Qty>
                <Pos:True>
                <F:Count>
                <Con:0>
            >
            <A:
                <N:5001>
                <AN:SV>
                <Att:EPOSSTransaction.SaleValue>
                <Pos:True>
                <F:Sum>
                <Con:0>
            >
        >
>
```

The following attributes are currently specified.

Field Name	Description	Format	Comments
Collection		String	EPOSSDNodes
ObjectName	Numeric Identifier identifying object node	Numeric	
StartDate	Effective Start Date		
EndDate	Effective End Date		
RData			
Data			
DN	Dynamic Name: Descriptive text name for this Node	Numeric	
GB	Group By: Identifies the fully qualified attribute this dynamic node will create groupings on (e.g.: EPOSSTransaction.SaleValue)	String	
GI	Group Interval: 'Every' indicates all items are grouped, a number indicates the items grouped before a page break occurs, used for pre-printed stationary reports such as giros. Repeating attribute describing all accumulators associated with this Node. In fact, no repeating version of this exist. The only values it takes are: Every, 16 or 23.	Numeric	
GSS	Direction of sorting (Forward)		
OB	Optional. Order By. This attribute is used when the ordering of a report is different from the Grouping, as specified by the GB attribute. If omitted, ordering is done by the attribute specified in the attribute GB. An example of this attribute is used in EPOSSDNode 5049s		

FUJ00085124

GST	Data type (Date; Number; String)		
GID	GroupID: this is also the ObjectName		
C	Optional Repeating Attribute, Child Node number: If this node has any children, these are listed using repeated occurrences of this attribute. If this node has no children then the attribute is not used.	Numeric	
A	Repeating group describing all accumulators associated with this node.		
N	The Node number is listed again when defining accumulates	Numeric	
AN	Accumulator Name: Defines what is to be accumulated; e.g. Record Count (RC), Sale Value (SV), Quantity (Qty)	String	SV, QTY, RC, SVETM, QTYETM, SVNTM, QTYNTM, LTSV
Att	Attribute: Identifies the fully qualified attribute this accumulator works on. Defines the attribute that is searched for within MessageStore (usually EPOSSTransaction.SaleValue or EPOSSTransaction.Qty)	String	EPOSSTransaction.SaleValue, EPOSSTransaction.Qty Alternative suffixes to EPOSSTransaction are RC, SI, LTSV, SVETM, QTYETM, SVNTM, QTYNTM
Pos	Positive Effect: Indicates if this accumulator totals in a positive or negative way. Used when a negative number (e.g. refund) is to be displayed as a positive number; if false is specified, the absolute value will be used	String	True or False
F	Function: Indicates how totals are derived using this accumulator. Defines whether the accumulated value should be summed or counted; e.g. the sale value (SV) should be summed to give total value of all transactions, the record count (RC) should be counted to give total number of transactions	String	Sum, Count, SumETM, SumNTM
Con	ContextID	Numeric	Usage Unknown

12.6 MessageDefs Collection

This collection holds definitions for each screen prompt – i.e. the narrative question, and any associated answer buttons and icons. See [RDREQ] for details. The following collection members are affected by CAP-TP transition. It should be noted that some of these messages are closely related to the actual production of the Cash Account. Since those may not be required when the branch is operating in TP mode for real, there seems little point in providing TP variants of them. However, it may be simpler to mindlessly provide TP variants of all of them then discard some later, rather than risk missing important cases.

FUJ00085124

In each case 2 new message variants will be provided with revised text and then the code will need to decide which one to use (dependent on CAP / TP mode). Following counter upgrade to S80 across the estate all of the existing messages can be removed.

Id	Ref	Caption	Text
40	MSG_SUDELETEHASBEENUSED	Could Not Delete	Could not delete stock unit %StockUnit% as it has been used in the current Cash Account Period.
58	MSG_ROLLINTONEXTCAP	Roll Over	Do you wish to roll over into the next Cash Account Period (%NextCAP%) or into the next Balance Period in this CAP (%ThisCAP%/%NextBP%)?
59	MSG_ROLLOVEROK	Rollover OK	Stock unit %StockUnit% has been successfully rolled over to Cash Account Period %CAP% Balance Period %BP%.
60	MSG_CONFIRMOFFICEROLLOVER	Confirm Rollover	WARNING - Check that all Office Accounting activities have taken place. Confirming will commence rolling over of the office into the next CAP.
63	MSG_ROLLCONFIRM	Confirm Rollover	Please confirm that you wish to rollover into the next BP (%NextBP%) in CAP (%ThisCAP%).‖You cannot roll into the next CAP (%NextCAP%) until the Cash Account is completed for the previous week
66	MSG_OFFICEROLLEDOVEROK	Rollover OK	Office Rollover to CAP %CAP% %Year% completed successfully.
77	MSG_REVERSALNOTCAP	Reversal Not Allowed	The transaction you attempted to reverse was not in the current Cash Account Period. Only transactions in the current Cash Account Period are allowed.
91	MSG_CREATEATCASHACCOUNT	Cannot Create	The balance option has been invoked. You cannot create stock units until the Cash Account is complete and the office has been rolled over to the next Cash Account Period.
136	MSG_ROLLNOOFFICEBALANCE	Cannot Roll Over	You have not yet produced a Cash Account report. The Cash Account report must be produced before the Office can be rolled over.
137	MSG_CASHACCOUNTNOOFFICEBALANCE	Cannot Produce	You cannot produce a Cash Account until the Office Balance Report has been confirmed.
139	MSG_ROLLAHEADOFOFFICE	Cannot Roll Over	Stock unit %StockUnit% is already in the next CAP (%NextCAP%). A stock unit cannot roll more than one CAP ahead of the office CAP (%CurrentCAP%).
189	MSG_OFFICECAP	Office CAP	The Office is currently in Cash Account Period %OfficeCAP%.>
193	MSG_INWRONGCAPLATE	CAP Passed	^Warning^ The expected end date for the CAP that SU %SU% is currently working in (%CAP%) has passed. Please double check you are in the correct CAP before starting work.>
194	MSG_INWRONGCAPEARLY	CAP Early	^Warning^ The expected start date for the CAP that SU %SU% is currently working in (%CAP%) is in the future. Please double check you are in the correct CAP before starting work.
196	MSG_CASHACCNOSNAPSHOT	No Snapshot	You cannot produce the Cash Account Report until you have printed or previewed the Cash Account Snapshot.
199	MSG_ACWRONGCAP	Invalid Date	This button is not available as today's date is not valid for the CAP. This stock unit is currently in Cash Account Period (%CAP%).
200	MSG_ROLLEARLY	Warning	^Warning^ The end date for CAP %CAP% is not today - please double check that you wish to roll over into the next CAP.
214	MSG_NEXTCAPGETDETAILSFAILED	Error	Could not retrieve details for the next CAP. Current CAP is %CAP%.
215	MSG_INWRONGCAPUNKNOWN	CAP Unknown	^Warning^ The expected start/end date for the CAP that SU %SU% is currently working in (%CAP%) could not be determined. Please double check you are in the correct CAP before starting work.
230	MSG_SUDELETENOTOFFICECAP	Could Not Delete	Could not delete stock unit %StockUnit% as it is not in the current Office CAP.
236	MSG_MANDATORY_SUSPENSE	Mandatory Report	You cannot produce a Cash Account until the Suspense Account Report has been produced.
241	MSG_DENYCAPEXT_NODATA	Cannot Extend	You cannot extend the rollover because there is no data present to support this request.

FUJ00085124

257	MSG_CASHACCOUNTERROR	System Error	An internal system error has occurred. Please re-run the Cash Account process.
261	MSG_MANDATORY_SUSPENSE_01	Mandatory Report	You cannot produce a Cash Account until the Suspense Account Report has been produced.
266	MSG_CONF_EXTENDEDCAP_YEAR	CAP Extended	CAP Rollover has been successfully extended from CAP %CurrentCAP% to CAP %ExtendedCAP%.
272	MSG_REVERSALTXNFORMATERROR	Transaction Not Found	The ID entered could not be found. Please check that the transaction ID is for the current CAP and that the format is correct.
300	MSG_NOOFFICEBALANCE	Cannot Continue	You have not yet confirmed the office balance. You must do this before you can produce the Cash Account.
335	MSG_DOESNOTBALANCE	Balancing Error	Receipts and Payments do not match, please investigate. The error may be able to be corrected using Reversal Functions. ^WARNING: Continuing may lead to an unbalanced Cash Account^
342	MSG_REVERSALTXNNOTEXIST	Transaction Not Found	The requested transaction was not found in the current CAP and cannot therefore be reversed.>
345	MSG_BESFB_INVALID_NEWDATE	Invalid Recovery Date	The date you have entered is invalid within the current CAP.‖Please touch the OK button to re-enter the date.
371	MSG_EXTENDED_2OR3_CAPPERIOD	Extended Period	Please select how many weeks you wish to extend the next rollover CAP
372	MSG_EXTENDED_1OR3_CAPPERIOD	Extended Period	Please select how many weeks you wish to extend the next rollover CAP
373	MSG_EXTENDED_1OR2_CAPPERIOD	Extended Period	Please select how many weeks you wish to extend the next rollover CAP
374	MSG_DENY_EXTENDEDCAP_YEAR	Cannot Extend	You cannot extend the rollover because the requested week is in a different cash accounting year.
375	MSG_DENY_EXTENDEDCAP_SU	Cannot Extend	You cannot extend the rollover because one or more of the Stock Units are in a different CAP to that of the Office CAP
441	MSG_APS_REVERSALNOTCAP	Reversal Not Allowed	The transaction you attempted to reverse was not in the current Cash Account Period. Only transactions in the current CAP can be reversed.
526	MSG_CONFIRMPRINTSOK	Confirm Printing	Touch Confirm to complete the office rollover. Touch Reprint to reprint the Cash Account.
527	MSG_RETRYORPREVIEW	Cash Account Report	
528	MSG_PREVIEWFINISHED	Confirm Preview	Ensure all details of the Cash Account have been noted
531	MSG_BESFB_INVALID_FALLBACK_DATE	Invalid PCHL Date	This help desk encashment cannot be recovered because the date in the corresponding PCHL transaction is invalid within the current CAP.‖Touch the OK button to continue.
537	MSG_PCDF_INVALID_CAP	Invalid CAP	Unable to report on the requested CAP. This may be because the CAP is an Extended CAP or is not a valid Office CAP.
538	MSG_PCDF_CAP_TOO_OLD	Invalid CAP	Unable to report on the requested CAP because it is too old.>
620	MSG_NEWSTOCKUNITDISCONNECTEDNEIGHBOURS	Disconnected counters	^Warning.^ Not all counters are connected. Creating a stock unit and performing transactions in it may lead to a receipts/payments imbalance if other counters are in higher office CAPs. Do you wish to continue?
621	MSG_WARN_EXT_CAP_DISCONNECT	Disconnected Counters	^WARNING^ There are currently one or more counters disconnected.‖Continuing may prevent you from producing your Cash Account.‖Do you wish to continue?
730	MSG_MANDATORY_MAILS_LABELS	Mandatory Report	You cannot produce a Cash Account until the Postage Labels Report has been produced.

12.7 ModeParameters Collection

The existing collection Modeparameters is impacted. New Mode instances will be defined to support the introduction of Corrections, and new attributes will be introduced to support the specification of Mode specific pricing. The latter change specifically applies to the need to

FUJ00085124

transact volume stock products at different prices depending upon the mode they are being transacted in.

Example:

```
<Collection:_ModeParameters>
<ObjectName:RIAD_06>
<StartDate:01-JAN-1996 01:01:31>
<EndDate:>
<RData:
        <ModeInfo:
                <IItem:IItem03197>
                <Cmd:ChangeMode>
                <DASS:True>
                <MaxStackTotal:9999999.99>
                <Mode:RIAD>
                <LINVZero:True>
                <MC:True>
                <SessionReceipt:107>
                <SettlementProduct:11215>
                <AlwaysPrintReceipt:True>
                <ReceiptTitle:Remittance In Slip (Auto Distribution)>
                <CallApp:>
                <ReceiptHotKey:Disabled>
                <ModeTitle:Rem In ADC>
                <ReverseSense:True>
                <PermanentSense:In>
                <PrimaryMappings:>
                <SecondaryMappings:
                        <L1:>
                        <L2:3047>
                        <L3:3028>
                        <L4:3027>
                        <L5:3017>
                <VolSValue:Zero>
        >
>
```

Field Name	Description	Format	Comments
Collection		String	ModeParameters
ObjectName	Identifier identifying object mode	String	
StartDate	Effective Start Date		
EndDate	Effective End Date		
RData			
ModeInfo			
IItem	Menu Hierarchy Desktop button reference for Mode	String	
Cmd	Command String invocation as a result of button depression	String	
DASS		Boolean	Unknown
MaxStackTotal	Total allowable stack total for transactions initiated in this Mode	Currency	

FUJ00085124

Mode	Mode Mnemonic, tallies to product allowablemodes in product reference data	String	
LINVZero		Boolean	Unknown
MC		Boolean	Unknown
SessionReceipt		Numeric	Unknown
SettlementProduct		String	
AlwaysPrintReceipt		Boolean	True or False
ReceiptTitle	Title to appear on receipt	String	
CallApp	Command String with which to invoke receipt printing	String	
ModeTitle			
ReverseSense			
PermanentSense			
PrimaryMappings			
SecondaryMappings			
ReceiptHotKey			
VolSValue	Value of product sale value to adopt when product transacted in this mode, which is denoted as Volume Stock	String	Sale, Loss or Zero allowed

12.8 BTSSourceData Collection

A new collection BTSSourceData is introduced. The collection provides a set of objects that soft map the Branch Trading Statement to the data which sources it. By being a soft mechanism for representing the BTS it also provides a means to drive changes to the report data content without necessarily requiring software changes.

Example:

```
<Collection:_BTSSourceData>
<ObjectName:LineCCC>
<StartDate:01-JAN-1996 00:00:31>
<EndDate:>
<RData:
        <Data:
                <RH:Cash On Hand>
                <AP:C>
                <SourceSUMethod:
                        <Method:SumAcc>
                        <Node:
                                <Id:1000>
                                <AccType:SVNTM>
                                <Effect:Neg>
                                <Operator:+>
                        >
                >
                <PrintMethod:
                        <Method:SummaryLine>
                        <ColTotal:>
```

FUJ00085124


```
                    <ColSuspense:
                        <PrintValue:False>
                    >
                    <ColSUs:
                        <PrintValue:True>
                        <CalculateMethod:Sum>
                        <Node:
                            <Type:Source>
                            <Operator:+>
                        >
                    >
                >
            >
>
```

The following attributes are currently specified.

Field Name	Description	Format	Comments
Collection		String	BTSSourceData
ObjectName	Mnemonic representing each data line on the report, structured to allow line inserts. A specific object 'Parameters' is used to define parameters that configure the use of rollover trailer identifiers and print control data	String	Eg LineDDD
StartDate	Effective Start Date		
EndDate	Effective End Date		
RData			
Data			
RH	Row Heading; the specific text heading each row. The declaration and signature text lines are 'specified' as row headings and hence the row heading can actually be up to 140 characters long	String	May have a value of 'White Space' denoting a blank heading. Alternatively attribute can be missing to result in a blank heading
AP	Accounting Period; specifies either the Current (C) or Current-1 (C-1) Trading Period. Data on the report is derived from both the current and last trading periods	String	C or C-1 are valid values
SourceSUMethod	Attribute Group describing any data for the line that has been sourced from Stock Unit Figures.		If not present no data is calculated nor recorded for the BTS. In such circumstances the line data can be derived entirely at the time of Report production
Method	Method name known locally to EPOSS StockUnit that allows parameters to be provided, enabling the BTS data for the current object line to be retrieved and stored for later use in BTS production		SumAcc, the one method currently defined
Node:	Repeat Data grouping, providing attributes to the Source Method		

FUJ00085124

Id:	The Node identifier from a data tree accumulation that provides the data	Numeric	
AccType:	The accumulator within the node which defines the specific node data	String	SV, QTY, RC, SVETM, QTYETM, RCETM, SVNTM, QTYNTM, RCNTM
Effect:	The accounting sense of the data item. Values will tend to be printed positive on the BTS but can be held negative. The Effect defines how the data is held and hence whether sign reversal is need. The source method will perform the sign reversal if required	String	
Operator	An operator (+ or -) defining the operation should the data item need to be combined with another	String	+ or -
Product:	Repeat Data grouping, providing attributes to the Source Method		
Id:	The Product identifier from a data tree accumulation that provides the data	Numeric	
AccType:	The accumulator within the node which defines the specific node data	String	SV, QTY, RC, SVETM, QTYETM, RCETM, SVNTM, QTYNTM, RCNTM
Effect:	The accounting sense of the data item. Values will tend to be printed positive on the BTS but can be held negative. The Effect defines how the data is held and hence whether sign reversal is need. The source method will perform the sign reversal if required	String	
Operator	An operator (+ or -) defining the operation should the data item need to be combined with another	String	+ or -
SourceSuspMethod	Attribute Group describing any data for the line that has been sourced from Office Figures.		If not present no data is calculated nor recorded for the BTS. In such circumstances the line data can be derived entirely at the time of Report production
Method	Method name known locally to EPOSS StockUnit that allows parameters to be provided, enabling the BTS data for the current object line to be retrieved and stored for later use in BTS production		SumAcc, the one method currently defined
Node:	Repeat Data grouping, providing attributes to the Source Method		

FUJ00085124

Id:	The Node identifier from a data tree accumulation that provides the data	Numeric	
AccType:	The accumulator within the node which defines the specific node data	String	SV, QTY, RC, SVETM, QTYETM, RCETM, SVNTM, QTYNTM, RCNTM
Effect:	The accounting sense of the data item. Values will tend to be printed positive on the BTS but can be held negative. The Effect defines how the data is held and hence whether sign reversal is need. The source method will perform the sign reversal if required	String	
Operator	An operator (+ or -) defining the operation should the data item need to be combined with another	String	+ or -
Product:	Repeat Data grouping, providing attributes to the Source Method		
Id:	The Product identifier from a data tree accumulation that provides the data	Numeric	
AccType:	The accumulator within the node which defines the specific node data	String	SV, QTY, RC, SVETM, QTYETM, RCETM, SVNTM, QTYNTM, RCNTM
Effect:	The accounting sense of the data item. Values will tend to be printed positive on the BTS but can be held negative. The Effect defines how the data is held and hence whether sign reversal is need. The source method will perform the sign reversal if required	String	
Operator	An operator (+ or -) defining the operation should the data item need to be combined with another	String	+ or -
PrintMethod	Attribute Group describing the data item to be printed		If not present then no value is printed
Method	Method name known locally to BESReports that allows parameters to be provided, enabling the BTS data for the current object line to be retrieved and placed on the BTS view. The following methods are defined SummaryLine - Print a Summary Page Line. The data derivation and makeup of the line are formulated from further parameters HeadingOnly - Just print Row Heading, which can be 'White Space' for a blank line StockHoldings - Prints the Stock Holdings		SummaryLine, HeadingOnly, StockHoldings, TransactionCorrections, NewPage

FUJ00085124

	Data Block, starting on a new page, includes headers and potentially scans multiple pages TransactionCorrrections - Prints the number of transaction corrections. Parameter <LineNumMoreThan:> may be used NewPage - Starts a new page with a BTS Page Header. Parameters <Node:>, <Type:> and <Source:> also used		
Line NumMoreThan	Used by <Method:NewPage>. Determines a conditional page throw is number of lines utilised on page exceeds this value	Numeric	Optional; if not present then page throw is unconditional
ColTotal	Used by <Type:SummaryLine>. A group determining the use of the branch total column. Determines whether a branch total column is to be printed. The current implementation determines the branch total as the sum of all printed line columns.		
PrintValue	Determines if the column is to be printed	Boolean	Default True
ColSuspense	Used by <Type:SummaryLine>. A group determining the use of the Suspense column. Determines whether a suspense column entry is to be printed and any derivation of the source data		
PrintValue	Determines if the column is to be printed	Boolean	Default True
CalculateMethod	Determines the data derivation calculation method. Currently the only implementation is Sum. All data items specified by the <Node:> parameter are summed	String	Sum
Node:	Repeat Data grouping, providing attributes to the Print Method		
Type	A submethod, or instruction within the method, identifying whether the node element is held locally, is sourced from a SourceMethod SumSourceSU - The sum of all externally derived SU entries on a named line. The line can be any line SumBTSLineSU - The sum of all already calculated SU entries on a named line. The line can be any line SourceSuspense - Derive the suspense item from an externally specified line item as specified. The line item must be alphabetically less than or equal to the current line BTSLineSuspense - Derive the suspense item from an item already calculated. The		SumSourceSU, BTSLineSuspense, SourceSuspense, SumBTSLineSU

FUJ00085124

	line item must be alphabetically less than or equal to the current line		
Id:	The BTS Line identifier sourcing this data item. If missing this means the line identifier is that of the current collection object	String	Eg. LINEDDD
Operator	An operator (+ or -) defining the operation should the data item need to be combined with another	String	+ or -
ColSUs	Used by <Type:SummaryLine>. A group determining the use of the Stock Units' columns. Determines whether stock unit column entries are to be printed and any derivation of the source data		
PrintValue	Determines if the column is to be printed	Boolean	Default True
CalculateMethod	Determines the data derivation calculation method. Currently the only implementation is Sum. All data items specified by the <Node:> parameter are summed	String	Sum
Node:	Repeat Data grouping, providing attributes to the Print Method		
Type	A submethod, or instruction within the method, identifying whether the node element is held locally, is sourced from a SourceMethod, or a combination of the two SourceSU - An externally derived SU entry on a named line. The line item must be alphabetically less than or equal to the current line BTSLineSU - Derive the item from an item already calculated. The line item must be alphabetically less than or equal to the current line		SourceSU, BTSLineSU
Id:	The BTS Line identifier sourcing this data item. If missing this means the line identifier is that of the current collection object	String	Eg. LINEDDD
Operator	An operator (+ or -) defining the operation should the data item need to be combined with another	String	+ or -
Print	Attribute Group describing print control parameters		Only applicable to object instance 'Parameters'
TotalLinesPerPage	Specifies the total number of lines that can be printed on any page	Numeric	44
SummaryLinesPerPge	Specifies the number of Summary Page Value lines to be printed on a page	Numeric	39
ProdsPerCol	Specifies the number of items to be reported per column on the Stock Holdings	Numeric	30

FUJ00085124

	Page		
ProdDescField	Specifies the product name to be used to identify the stock holding	String	Data.RN
EndofReport	Provides the End of Report Trailer Text, automatically output, centred	String	*** END OF REPORT ***
IncludeValueStock	Determines whether value stock items are to be included in the stock holdings figures on the stock holdings page	Boolean	False
C	Attribute Group describing the current AP rollover trailer identifiers	String	Only applicable to object instance 'Parameters'
Previous	Currently unused	String	
Current	Provides the rollover trailer linking all BTS Items that relate to the current TP	String	BTSCFFiguresTrailer
C-1	Attribute Group describing the previous AP rollover trailer identifiers	String	Only applicable to object instance 'Parameters'
Previous	Provides the name of the rollover trailer used to record private trailer figures in the previous TP, as Brought Forward figures in the current TP. It is this rollover trailer that is replaced by C-1:Current	String	BTSBFFiguresTrailer
Current	Provides the rollover trailer linking all BTS Items that relate to the previous TP	String	CurrentBTSBFFiguresTrailer

12.9 SuspenseSections Collection

A new collection SuspenseSections is introduced. The collection provides a set of objects that soft map the report groups for the suspense report and cash in pouches report to the data which sources the content of each section. The collection replaces the previously used SuspenseGroups collection.

Example:

```
<Collection:_SuspenseSections>
<ObjectName:SectionAA>
<StartDate:01-JAN-1996 00:00:31>
<EndDate:>
<RData:
        <Data:
                <GN:Cash In Pouches>
                <PID:5610>
                <ShowMovements:False>
                <DetailMsg:  WARNING - Check this C/Fwd column equals the actual total value of pouches>
                <DetailMsg:   awaiting collection. If it does not, print the Cash In Pouches Awaiting Collection>
                <DetailMsg:   report to establish where the discrepancy is.>
                <CAMappings:5075>
        >
>
```

The following attributes are currently specified.

Field Name	Description	Format	Comments

FUJ00085124

Collection		String	SuspenseSections
ObjectName	Mnemonic representing each data group on the reports, named so that ascending object names represent the group ordering on the reports and structured to allow line inserts.	String	Eg SectionBBB, allows a group SectionBBA to be inserted before it
StartDate	Effective Start Date		
EndDate	Effective End Date		
RData			
Data			
GN	Group Name, provides the legend heading leading each group on the reports	String	The Cash In Pouches Report only contains one group
PID	Repeating attribute providing the product number of each product, transactions of which source this suspense group	Numeric	
ShowMovements	Indicator providing facility to summarise group data into a subtotal rather than show each movement. Used on the suspense report to summarise cash in pouch data rather than report each movement. Movement data for cash in pouches is reported on the Cash In Pouches Report	Boolean	True or False. Default value is True
DetailMsg	Repeating attribute providing a message that is output when movements reporting is inhibited	String	
CAMappings	Provides the old link to any migrated brought forward figures to the reports. The possibility of a new office having suspense movements migrated prior to point 20 in the Impact migration process needs to be taken account of. Previously this was catered for by it being the object reference, which also stipulated the order of the groups on the report	Numeric	

FUJ00085124

13.0 Appendix C – Affected Persistent Object Collections

This appendix provides a reference to the affected persistent object collections as a result of changes to Balancing, rollover and handling stock by volume at S80. Each collection is addressed by a separate sub section.

13.1 EPOSSCAP Collection

This collection contains a set of specific persistent objects, maintained by EPOSS and LFS to record the state of the Office.

The collection currently contains two objects 'LFS' and 'Office'. 'LFS' is not relevant to this design so is not discussed further. The 'Office' object describes the current Cash Account Period (CAP) for the Office, is written after office rollover, and has the following syntax

<Collection:EPOSSCAP>	
<ObjectName:Office>	
<Data:	
<MailsLabelsPrinted:<CAP14:TRUE>>	Postage labels report has been produced prior to rollover
<SURAT:CI4>	Non-value stock has been declared prior to rollover. (This becomes obsolete once transition to TP mode is complete – see [DP] section 2.5.1.4.2).
<SuspenseAccountPrinted:<CAP14:TRUE>>	Suspense account report has been produced prior to rollover
<CAPTrailer:<GroupId:901777><Id:1><Num:37141>>	Identifies the CAP trailer from the last CAP rollover
<RolloverTrailer:<GroupId:901777><Id:1><Num:37096>>	Identifies the Rollover trailer from the last CAP rollover
<PreviousCAP:13>	Previous CAP/TP §
<CAP:14>	Current CAP/TP §
<NextCAP:15>	Next CAP/TP §
<Year:2004>	Financial year for current CAP/TP §
<StartDate:24-JUN-2004 00:00:00>	Start date for current CAP/TP §
<EndDate:30-JUN-2004 23:59:59>	End date for current CAP/TP §
<PrevWeekEnd:23-JUN-2004 23:59:59>	End date for previous CAP/TP §
<TP:1>	These describe the transition states when moving from CAP to TP-based accounting. See Final CAP Rollover and Transition (5.1.2.6).
<TPTransition:1>	
>	

Figure 13 - EPOSSCAP Office Object Syntax

§ Note that for backwards compatibility, CAP-related fields are reused to hold the equivalent TP information where practical.. See Approach Taken to Transition (5.1.2.3) for more details.

13.2 StockUnits Collection

This collection holds all the Stock Units with their state.

<Collection:StockUnits>

FUJ00085124

<ObjectName:SU1>	Name of the Stock Unit
<Data:	
<Shared:False>	True for shared or False for individual stock units.
<Locked:False>	True if locked, False otherwise.
<LockedBy:>	Name of locking User (if any)
<CAP:14>	Current CAP/TP §
<BP:1>	Current Balancing Period
<NextCAP:15>	Next CAP/TP §
<StartDate: 01/07/2004>	Starting Date for current CAP/TP § (dd/mm/yy)
<EndDate: 07/07/2004 23:59:59>	Ending Date for current CAP/TP § (dd/mm/yy hh:mm:ss)
<UserBalancing:MIGR01>	User Id balancing the stock unit
<Inactive:False>	True if the inactive, False otherwise
<Balancing:False>	True if in the process of balancing, False otherwise
<BalanceStatus:Dirty>	Dirty if transactions performed, Clean otherwise
<RolloverTrailer: <GroupId:901777><Id:1><Num:59098>>	Points to the Rollover Trailer message
<CAPRolloverTrailer:<GroupId:901777><Id:1><Num:59053>	Points to the CAP Rollover Trailer message
<AdjustStock:True>	Denotes whether stock has been adjusted
<TP:1> <TPTransition:1>	These describe the transition states when moving from CAP to TP-based accounting. See Final CAP Rollover and Transition (5.1.2.6)
>	

Figure 14 - StockUnits Object Syntax

FUJ00085124

14.0 Appendix D – Overview of Riposte Message Expiry

This appendix provides an overview of the coordinated management of Message Store Messages by both counter applications and Riposte with respect to recording Message Expiry values.

Each message in the message store contains a number of attributes. One such attribute is the <Expiry:> attribute. This attribute, the value of which is held as a number of days, determines the period for which the message will remain in the message store, after which it is deleted by Riposte archiving software. In relation to Extending Retention Periods all messages that are required for the EPOSS Balancing functionality must be retained long enough for them to be still available when the balancing process commences, and throughout the balancing process if their need is still required. . In addition, all messages that are required to support reporting and other transactional activity within an extended Trading Period also need to be retained up to the point of successful Trading Period rollover.

When a message is written to the message store by a counter application it may or may not provide a value for the <Expiry:> attribute to the message. If no value is supplied Riposte provides a value, being that of the 'DefaultMessageExpiry' from Riposte Configuration Parameters.

If the application provides a value then Riposte will determine if the value is within the limits of the 'MinMessageExpiry' and the 'MaxMessageExpiry' within Riposte Configuration Parameters. If the supplied value is outside the range then it is replaced by the appropriate limit to enforce a value within the range.

FUJ00085124

15.0 Appendix E – EPOSS Accounting Node Structure

The following appendix provides an introduction to the Accounting Node Structure.

A major part of Report Preparation involves the initialisation and supply of the parameters for EPOSS Data Server. These parameters relate to the EPOSS Accounting Node Structure, and establish how an instance of the structure will be built for a particular report given the contributing transactions. In other words, the Reporting Application acts as a client Application and Data Server as a Server component.

The Accounting Node Structure is built by Data Server based on a structure specified in reference data by the collections EPOSSNodes and EPOSSDNodes. Through EPOSSNodes an Accounting Hierarchy is specified forming the summarisation hierarchy for the accumulation of transaction quantities and values in a report. The Dynamic Node structure, through EPOSSDNodes, determines the way in which the Transaction Data is structured so that information of like transactions can also be summarised. The dynamic structure can be used as a vehicle to order report content, for example by Stock Unit within Balance Period.

The EPOSS Accounting Hierarchy consists of a structure of EPOSS Accounting Nodes. The accounting node hierarchy is supplied by Fujitsu Services and agreed with Post Office Limited and is made up of a selection of Nodes which are used to accumulate product transactions in each of the different modes (e.g. Serve Customer, Remittance). These nodes belong to a collection EPOSSNodes and their position in the hierarchy determine the level of abstraction at which a product is described and their level of accumulation. Each Product when transacted in serve customer mode is given a set of primary mappings, (defined in EPOSSProducts), and transactions may also be given a secondary mapping depending on the mode in which they are transacted. Secondary mappings are optionally defined against each mode in collection ModeParameters, this secondary mapping is defaulted to all transactions performed in a particular mode. The Secondary mapping may however be overridden for specific product transactions if an entry for that product/mode combination exists in collection CofAProductModes. This collection will be replaced at Impact Release 3 by the use of Post Office Product Modes. A Stock Product that is analysed by Volume as opposed to Value also has Tertiary Mapping, these being defined within the EPOSSProducts collection optionally and will be applied to all transactions of products for which Tertiary Mappings are defined, given the mode of the transaction.

These mappings together tell the system where on the node hierarchy to accumulate each transaction for each of the Primary, Secondary and Tertiary mappings.

Therefore:

- All transactions have a Primary Mapping
- Some transactions have a secondary mapping (depending on Mode of transaction)
- Some transactions have a Tertiary mapping (depending on the product being transacted and the Mode of the transaction)

The Node levels are structured in the EPOSS Accounting hierarchy in a decreasing levels, i.e. the Level 5 Node is at the top most level of the Accounting Hierarchy and the Level 1 Node is at the lowest level of the Hierarchy.

FUJ00085124

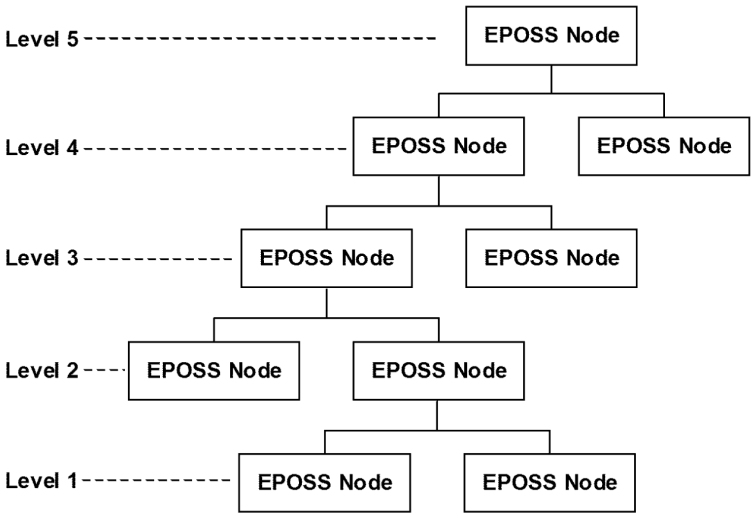


Figure 15 – EPOSS Accounting Node Hierarchy

These levels will be populated with different EPOSSNodes, each describing a product at a different level of abstraction. The Level 1 Node is the lowest level of abstraction, e.g. "First Class Stamp", and the Level 2 Node would be "Postage Item". There is a parent-child relationship, where the "1st Class Stamp" is a child node of the "Postage Item". The specification of which Accounting Nodes will be accumulated into, when a particular product is transacted, is specified by Reference Data, in the Product Collection.

The First Class Stamp, for example, has a EPOSSNode mapping of five levels specified, by the attributes <L1:1704><L2:2055><L3:3007><L4:3008><L5:3017> as shown below:

```
<Collection:EPOSSProducts>
<ObjectName:19>
    :
    :
    :
        <PM:
            <L1:1704>
            <L2:2055>
            <L3:3007>
            <L4:3008>
            <L5:3017>>
    :
    :
    :
```

Figure 16 – Example EPOSSNode Primary Mapping

Each EPOSSNode currently has three associated data leaves, or accumulators. These accumulators define what data is to be accumulated from each transaction that contributes to the node. The main accumulators have one data leaf relating to the Quantity (Qty), the second Record Count (RC), providing a count of transactions contributing to the node, and the third relates to Sales Value (SV). The data leaves, or accumulators, exist as a combination of

FUJ00085124

specifying the transaction attribute to be aggregated, together with a function describing how the data is to be aggregated. Data attributes can be summed (Sum) or counted (Count). So to obtain a 'count' of transactions the accumulator RC is incremented by 1 for each transaction. However counting sale values would give an incorrect result; rather sale values are summed.

By extending the potential available accumulators to include additional Functions it is possible to add conditional functionality to Data Server without incorporating hard coded Business rules within the Server component.

If a transaction involved the sales of a number of First Class Stamps, the Product Specification (EPOSSProduct) for the item will specify which nodes on the hierarchy the quantities and values are to be accumulated into. The accumulators will be accumulated accordingly, so that the specified nodes in the Accounting Hierarchy will be updated with the transaction data. This means that in the case of the First Class Stamp, the parent Node above "Postage Item" will also be updated when the transaction values are accumulated. This accumulation will proceed up the hierarchy of Nodes reaching the Balancing Node (3017), as depicted in the following diagram.

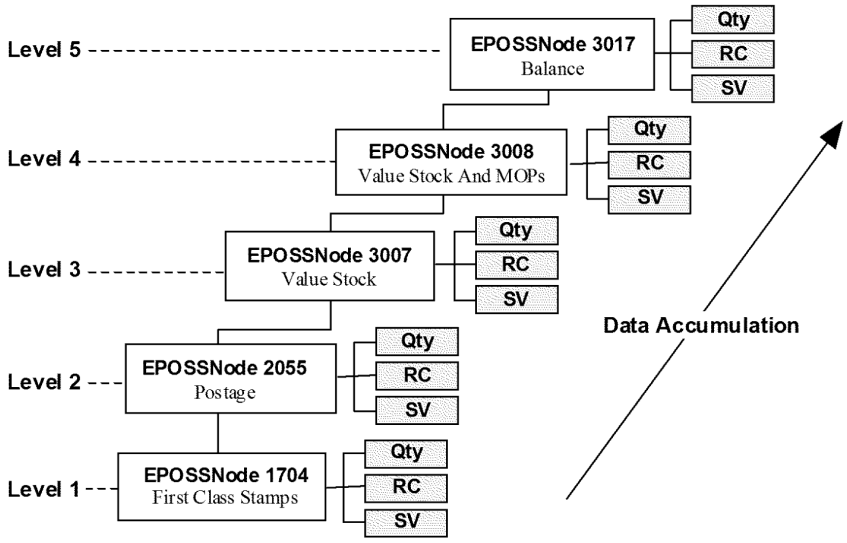


Figure 17 – EPOSSNode Data Accumulation

The parent and child relationship defines the hierarchy between Group Nodes, where a parent Node with an accumulate item is updated whenever the child Group Node accumulate item is updated.

In addition to building the Accounting Node Hierarchy, Data Server also builds the EPOSS Dynamic Node (EPOSSDNode) chain specification which determines how the Reports are grouped and ordered. EPOSSDNodes form a hierarchy structured according to the required data groupings to appear on the report appended to the last node in the hierarchy. Although the figure above shows a Level 1 Node as the last, in some transactions the hierarchy may end at a Level 2 Node. The DNode specification is not dependent on the level at which the Node structure ends, as the DNodes are appended to the terminating Node.

The Dynamic Node specification for a report may, for example, dictate that the transactions will be grouped first by Session Date, Stock Unit, Balance Period and finally by Session

FUJ00085124

Identifiers. The EPOSS Accounting Node Structure, which consists of both the static Accounting Nodes and dynamic DNodes, is a means for populating and ordering the data for the production of the Report. Data Server builds the structure based on the EPOSSNode and EPOSSDNode specification and accumulates the data leaves accordingly with the transaction Quantities, Record Counts and Sale Values. The structure then provides all the necessary data for populating the Report. The full structure is shown below.

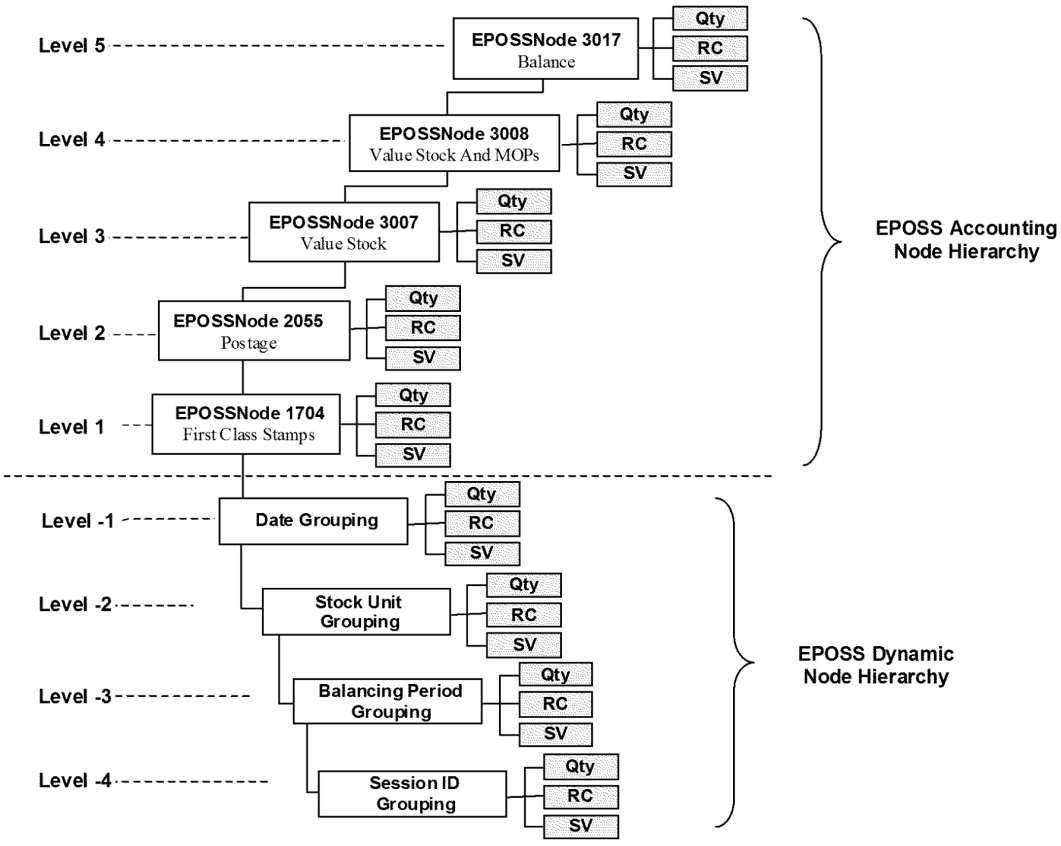


Figure 18 – Dynamic EPOSS Nodes

Therefore, report specific dynamic accounting nodes may be defined over and above the Accounting Hierarchy. Quantity and value totals plus transaction counts are also accumulated for the Dynamic Nodes.

The Dynamic Nodes provide for the accumulation of data in specific groupings, e.g. a dynamic node may group all products being reported by date or by stock unit or by date within stock unit. The dynamic nodes effectively provide the sorting and sub-totalling facilities.

The structure is also populated by Data Server with the selected transactions messages. These transaction messages are extracted from the Message Store, based on the Report Criteria.

The input to Data Server from the Reporting Service, would be the Report Criteria string, the EPOSSNode and EPOSSDNode specifications for that particular report. Data Server uses this information, extracts the transaction messages based on the Report Criteria. Using SQL-based

FUJ00085124

queries built from the Report Criteria, Data Server is returned a record containing the subset of transaction messages from the RMS.

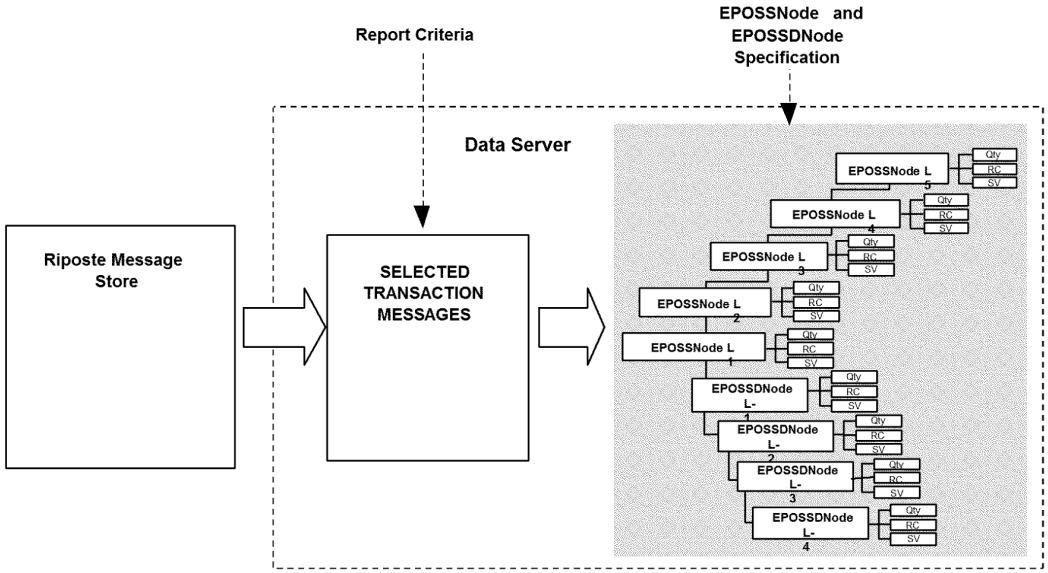


Figure 19 – Populating the EPOSSNode Tree

When the transaction messages have been extracted, Data Server builds an instance of the structure. This is based on the EPOSSNode and EPOSSDNode specification. This structure may be likened to a tree structure comprising the empty data leaves. The transaction data is then accumulated throughout the structure. When this task is completed, the structure is made available to the Reporting Client. An example of an actual instance of a structure is shown below.

FUJ00085124

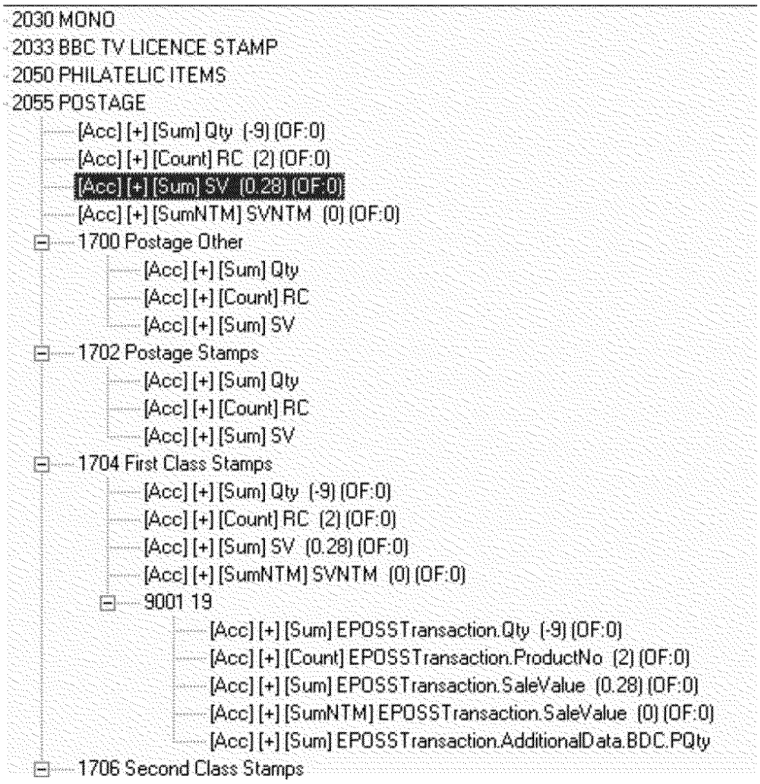


Figure 20 – Sample Populated EPOSSNode Tree

The Reporting Service subsequently extracts the Data from the structure as required, when the Report is printed or previewed.