FUJ00117516

Document Title: PAS/CMS MIS Data Extract HLD

Document Type: High Level Design and Interface Document

Abstract: A description of the ICL Pathway data extract from PAS/CMS to provide information to the MIS data warehouse to satisfy the requirements of release 2.

Status: Issued

Distribution:

W Hillyard	G Jackson	P Sewell
G Lloyd	A Ward	RC Smith (ESC/ITX)
D Brady (CFM)	S Channel	G Jenkins
K Linfoot (Oracle)	C Oudot (PDA)	
J Holmes	B Taylor	
J Hunt	G Boyce (CFM)	
P Lindsay	R Long	

Author: Pete Jobson (ICL Pathway)

Approval Authority: Graham Lloyd

Signature/Date:

Quality Authority:

Signature/Date:

Programme Delivery Authority:

Signature/Date:

FUJ00117516

0 DOCUMENT CONTROL

0.1 DOCUMENT HISTORY

Version	Date	Reason
0.1	**04/07/96**	Preliminary draft for discussion
0.2	**3/11/97**	Amended in accordance with review comments
0.3	**29/12/97**	Changes for PAS/CMS release 203
0.4	**25/02/98**	Changes for CP903
1.0	**02/04/98**	Issued

0.2 ASSOCIATED DOCUMENTS

Ref.	Version	Date	Title	Source
1 - HLFD\IFD\0006	6.02a	27/03/97	CAPS to PAS/CMS Data Interface Definitions and Validation Rules R3	CAPS
2 - SU/DES/0001	6.0	25/06/97	CAPS Access Service HLD	Pathway
3 - SU/DES/0008	2.0	09/07/97	CAPS Access Service (On-line) HLD	Pathway
4 - CG-DSS01/HDES/001	4a	31/07/97	High Level Design Report - Pathway Payment Authorisation Service and Card Management Service	Oracle
5 - DW/INF/000x	0.3	??	MIS Release 2: CMS/PAS EPID	CFM
6	4		BA/POCL MIS Requirements	PDA
7 - CR/FSP/0004	2.1	20/03/97	Service Architecture Design Document	Pathway
8 -			Schedule D08 - PAS Service Levels	PDA
9 -			Schedule E08 - CMS Service Levels	PDA
10 - DW/REQ/0005	0.4	17/02/97	Data Warehouse FRMS Requirements	Pathway
11 - SCP593		09/06/97	PAS/CMS Support For Schedules D08 & E08	Oracle
12 - SCP686		22/07/97	MIS Priority 2 Requirements for Release 2	Oracle
13 -	0.2	25/08/97	MIS Release 2: Scope of Implementation	CFM
14 - CR/FSP/011	0.2	23/07/97	Reconciliation of Cards & Temporary Tokens	Pathway
15 -			Designer 2000 - PAS/CMS	Oracle

FUJ00117516

0.3 ABBREVIATIONS

CAS	CAPS Access Service
CMS	Card Management Service
DLO	Dead Letter Office
ICMF	Inward Control Matching File
NPO	Nominated Post Office
OCMF	Outward Control Matching File
PAS	Payment Authorisation Service
PUN(S)	Pick-up Notice(s)
RPOI	Restricted Post Office Indicator
SLA	Service Level Agreement

FUJ00117516

0.4 Expected Changes

FUJ00117516

0.5 TABLE OF CONTENT

FUJ00117516

FUJ00117516

FUJ00117516

FUJ00117516

1 Introduction

In order to enable the monitoring of SLAs, the production of invoices, the evaluation of fraud risk and other MIS reporting, the data warehouse requires data sourced from the PAS/CMS system. This document provides a high level design of PAS/CMS modules required to support the system interfaces between PAS/CMS and the data warehouse.

This document intends to define the nature of the data interface, the scope of the interface and the responsibilities of the interfacing parties.

1.1 Scope

The scope of this document is contained to R2 and the requirements stated in the associated documents.

This interface document describes the following data:

- The location of the source data as it resides in PAS/CMS
- The nature and format of the flat file data as it will be provided to the data warehouse
- The structure and content of data required by the extracting processes for both the purposes of reference and temporary consolidation

In addition, it will describe the following processes:

- The processes which shall perform the data extract, their timeliness and their dependencies on the processes which produce and maintain the source data.
- It will define the responsibilities of the PAS/CMS system to ensure that the data is available in an accessible format
- The Maestro Schedule and Dependencies
- The level of Resilience & Recovery

This document will not include detailed module design nor will it describe the operational procedures necessary to execute the processes.

2

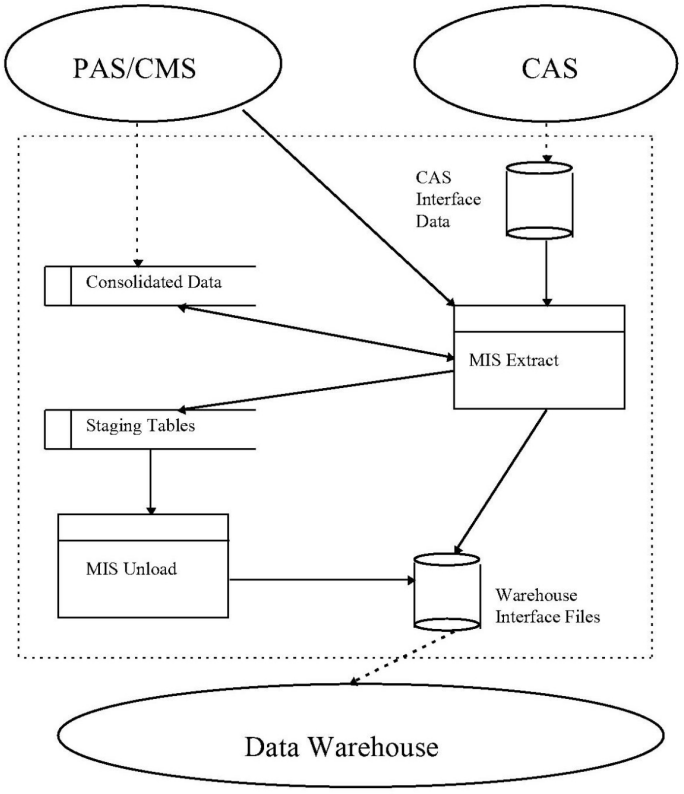
FUJ00117516

Architecture

2.1 Context and Responsibility

Both the processes and the data described within this document shall reside within the PAS/CMS domain and schema. As such, the MIS Extract is an integrated part of PAS/CMS. However, from a contractual and development ownership perspective, boundaries of responsibility need to be drawn.

The diagram below indicates the scope of implementation and the area of responsibility of the MIS Extract in the context of both the PAS/CMS and Data Warehouse systems. The area within the dotted boundary is deemed to be the responsibility of the MIS Extract with the exception of direct data feeds across the boundary which shall be the responsibility of the source systems (as shown by the dotted feeds).



FUJ00117516

2.2 Interface Functional Components

2.2.1 PAS/CMS

The PAS/CMS system encompasses the production, distribution and management of benefit collection cards as well as the processing of payment authorisations and their subsequent encashment. It is the activity of this system that the MIS Extract is responsible for recording and providing to the Data Warehouse for later reporting.

2.2.2 CAS

CAS is the interface between PAS/CMS and the Benefits Agency systems. CAS resides on both the Sequent and on the same platform as the Benefits Agency systems and there will therefore be one instance of CAS per Benefits Agency machine. Since CAS provides the boundary of responsibility between ICL Pathway and the Benefits Agency, it is at this point that much of the SLA information is captured.

2.2.3 MIS Extract

The Data Warehouse requires summarised, consolidated and aggregated information from PAS/CMS. The format and layout of this data is pre-defined and static. The first phase of the MIS Extract is to extract/aggregate information from the source tables. This part of the MIS Extract has the highest priority since much of the data to be extracted is only available for a brief period before being purged by PAS/CMS ready for the next on-line day. The eventual goal of the extract process will be to produce a set of flat files which are required by the Data Warehouse or a set of staging tables which adhere to the same structure as those flat files.

2.2.4 MIS Unload

Initiated only after all the staging tables have been populated for the day, the MIS unload is responsible for downloading the consolidated data into flat files for delivery to the Data Warehouse

2.2.5 Data Warehouse

The Data Warehouse is responsible for the detection of delivery of the flat file interface information and the subsequent upload of that information into the Data Warehouse database. The Data Warehouse is also responsible for the housekeeping of the flat files and their associated staging area.

2.3 Interface Data

2.3.1 Event Driven MIS Data

PAS/CMS is primarily concerned and optimised to perform the tasks outlined in section 2.2.1 above. As a result, the PAS/CMS schema is not always in an optimum form for the MIS Extract. Where these instances have been identified, it has been

FUJ00117516

agreed that PAS/CMS processes will identify MIS Extract data during the course of daily processing and feed this information directly into MIS Extract tables.

PAS/CMS shall be responsible for the integrity and validation of the information provided in this manner. However, the content of these tables is under the ownership of the MIS Extract which will be responsible for the housekeeping of such data.

2.3.2 CAS Interface Data

CAS (VME) logs all information regarding the transmission of files and transactions between the Benefits Agency (CAPS) and PAS/CMS on Inward and Outward Control Logs. This data is accumulated on the VME machine and is transferred regularly to the Sequent SMP in the form of serial files: ICMF (Inward Control Matching File), and OCMF (Outward Control Matching File) which are described in Ref: [1]. This data is used by the MIS Extract to supply the Data Warehouse with information to support the SLA's in regard to the timeliness of data delivery across the PAS/CMS - CAPS interface. CAS shall be responsible for the integrity and validity of this information.

2.3.3 Intermediate Data & Worksheet Data

The MIS Extract will place prime importance on the extraction of information from PAS/CMS where such source information is transitory. This information may be summarised into an intermediate stage prior to formatting into a form acceptable to the Data Warehouse.

The MIS Extract will maintain its own working data relating to the management of its own operations. Intermediate and worksheet data will reside within the PAS/CMS schema.

2.3.4 Staging Tables

A set of tables will be created on a daily basis which will mirror the information that will be required within the flat files required by the Data Warehouse. These will be transient and will be removed once the flat files have been created and backed-up. The staging tables will reside within the PAS/CMS schema.

2.3.5 Data Warehouse Flat Files

The data will be delivered to the Data Warehouse in the form of flat files. The MIS Extract will be responsible for ensuring that these files conform to the requirements of the Data Warehouse in both form and content. It will be the responsibility of the Data Warehouse to upload this data into the Warehouse Database and to perform the housekeeping of both the flat files and the flat file unload area.

2.3.6 Data Warehouse Interface

The Data Warehouse interface will be implemented using an NFS partition under UNIX. The data warehouse will be responsible for the creation of directories within which the MIS Extract will place the interface files.

The MIS Extract will be responsible for creation of interface files and a control file in the format laid down in reference [5] after which, a lock file will be created signifying completion of the extract. After creation of the lock file, it is assumed that the data

FUJ00117516

warehouse has received the extract. Refer to reference [5] for full details of interface, control and lock file formats.

The housekeeping of the NFS Partition is the responsibility of the Data Warehouse.

FUJ00117516

3 Resilience

MIS Extract processes can be split into two types; those which extract directly to the Data Warehouse interface files and those that extract via staging tables.

3.1 Direct Extracts

All the direct extract processes may be re-run on abnormal end and will re-create the Warehouse interface file from scratch.

In the event of loss of the Warehouse interface file after the extract has completed, each of the direct extracts may be re-run for the necessary date to re-create the Interface file. Manual intervention will be required to ensure that the repository directory exists and to ensure that the interface file is uploaded to the Warehouse database.

Re-running extracts for previous days is dependent on the information being available in the PAS/CMS database. This is individually described against each of the extract processes in section 0 - 6.3 Direct Extracts.

3.2 Two-Phase Extracts

Two-Phase extracts extract data from the PAS/CMS database and place the resultant data into a set of staging tables which exist for the date of the extract. This strategy is applied where the source data exists for only a short period so that it may be re-supplied to the Data Warehouse in the event of data loss. Eventually these staging tables will be truncated and dropped.

All two-phase processes may be re-run on abend. However, since the data being extracted is in the necessary state for only a short period, the extract must be re-run prior to executing any PAS/CMS process which may affect that source data. Further information on the transient nature of the PAS/CMS data may be found in section 0 - 7.1 Source Data and against each process definition in section 0 -6.4 Two-Phase Extracts.

Once the data is held within the staging tables, it may be re-extracted at any time until such data is purged.

FUJ00117516

4 Extract Schedule

The MIS Extract will extract data from PAS/CMS and provide a full set of data to the Data Warehouse by 02:00 on a daily basis.

All of the MIS Extract processes are dependent on the source information being available at the time of execution. Due to the large volumes of information passing through PAS/CMS on a daily basis, much of the source information is transient. MIS Extract processes will therefore need to knit closely with the PAS/CMS schedule to ensure that the information required is both available and complete at the time of the extract. Due to these dependencies, the MIS Extract cannot therefore commit to providing all of the information required by the Data Warehouse by a specific time. Some of the data files to be extracted will therefore always be delivered one day in arrears and these will be identified in the detailed process specifications in section 0 - 6 Application Processes.

4.1 Maestro

All of the MIS Extract processes will be scheduled and executed by Maestro within the PAS/CMS schedule. The extract process for each day will begin when a new PAS/CMS day is deemed to start. This is indicated by a change of system date which is instigated by PAS/CMS process SUPC702.

Apart from dependencies on both PAS/CMS processes and other MIS Extract processes, most of the MIS Extract jobs may execute either during the POCL Core Day or within the Overnight Schedule.

FUJ00117516

4.2 Process Names & Dependencies

Below is a table of MIS Extract processes and their dependencies. More detailed descriptions of each module and their dependencies may be found in section 0 - 6 Application Processes and section 0 - 7.1 Source Data.

Name	Cycle	Description	Reference	Dependencies
MISC101	Daily	Begin a new extract day	0	After SUPC702
MISC102	Daily	Deliver final information to the Data Warehouse	0	After all MIS Processes except MISC103
MISC103	Daily	Purge MIS Data	0	After MISC102
MISC104	Adhoc	Fallback Delivery	0	Anytime
MISC201	Daily	Help Desk Calls	0	***At*** 20:00 and prior to SUPC313
MISC202	Daily	Reference Data	0	After MISC101
MISC203	Daily	On-line Interface Transactions	0	After MISC101
MISC204	Daily	PO Closure Details	0	After MISC101
MISC205	Daily	Changes of NPO not Reversed	0	After MISC101
MISC206	Daily	Temporary Token Details	0	After MISC101
MISC207	Daily	Notifications of PUNS Not Received	0	After MISC101
MISC208	Daily	Card Production Data	0	After MISC101
MISC209	Daily	Random Encashment Information	0	At 18:00
MISC301	Weekly	Ongoing Active Cards	0	After CMSC109
MISC302	Daily	Cards Issued per Post Office	0	After MISC101
MISC303	Weekly	Cards Deactivated per Post Office	0	After CMSC115
MISC304	Daily	Card Event Details	0	After CMSC115
MISC305	Daily	Benefit Non-Encashment	0	After MISC101
MISC306	Daily	Encashments Made After Stop Received	0	After MISC101
MISC307	Daily	Customers With Many Changes of NPO	0	After MISC101
MISC308	Monthly	New Cardholders	6.4.8	After CMSC201 (non-urgent) and CMSC115
MISC309	Daily	Temporary Tokens Out of Stock	6.4.9	After CMSC401
MISC401	Monthly	Summary Data	0	After MISC101
MISC402	Daily	Batch Timeliness CAPS to PAS	0	At 20:00
MISC403	Daily	Batch Acceptance/Rejection & Batch Timeliness PAS to CAPS	0 & 0	After MISC101

FUJ00117516

MISC404	Monthly	Cardholder End of Interest	6.5.7	After CMSC109

5 Performance

Each MIS Extract process has been evaluated independently. Where data collection of source data by MIS Processes appeared too onerous, such data has been extracted on an event driven basis during daily PAS/CMS processing.

Within the release 2 MIS Extract, the responsibility of Data Warehouse Interface File production has moved within the scope of the Extract. This enables data to be Extracted directly to the Interface files without the need for intermediate storage. This reduces the need to move data between temporary stores in those areas where the data is recoverable at a later stage.

Each MIS Extract process will be evaluated as a candidate for parallel queries and optimiser hints. The number of parallel processes for candidate processes will be determined by an entry in the PAS/CMS system parameters although parallel processing will only be utilised during the overnight schedule. Those processes which are candidates for parallel queries may be switched between parallel and non-parallel to retain flexibility within the schedule.

FUJ00117516

6 Application Processes

Each MIS Extract process is described below in more detail than would normally be expected in a high level design document. Although this may be too detailed for the casual reader who requires a brief overview, it serves the following purposes:

- To ensure that the assumptions made about the availability of data and dependencies within PAS/CMS may be reviewed
- To ensure a common understanding about the data to be provided to the Data Warehouse
- To provide a document in a format detailed enough to allow test scripts to be compiled

Each of the processes described refers to data which resides in one of the following general areas:

Area	Description	Described Within
PAS/CMS	Tables residing within the domain of PAS/CMS operational system	0 - 7.1 Source Data Reference [4] Designer 2000 7.1 Source Data
Meta Data	Describing MIS Extract Data	0 - 7.2 MIS Extract Meta-Data
Intermediate Data	Temporary storage of extract results	0 - 7.3 Intermediate Data
Staging Tables	A set of 10 tables which are created at the start of the MIS Extract Day. They are dropped some time later by the MIS Purge process	0 - 7.4 Staging Tables
Interface Files	The actual interface files which are handed to the data warehouse	0 - 7.5 Warehouse Interface Files Reference [5]

6.1 General

6.1.1 Process Audit Trails

All MIS Extract processes will conform to the PAS/CMS method of writing Process Audit Trails as described in [**4**]. Mandatory items to be populated are:

- Sequence No
- Process Item - which must be included during Data Warehouse Interface File creation
- Timestamp
- Elements - Indicating number of rows inserted/created where applicable

FUJ00117516

- PID - UNIX Process Id
- Module Id

6.1.2 Error & Exception Handling

All MIS Extract processes will conform to the PAS/CMS method of writing Error Reports as described in [**4**]. However, since application exceptions will terminate any MIS process in its entirety, these exceptions will be treated in the same manner as Oracle/System errors. The errors will be written to the log filename which is passed as the last parameter to the process (default=stdout) and will include:

- Module name
- Timestamp
- Message Text - Oracle error number/text and/or relevant information determined by the process

6.1.3 Daily Files Update

A row is inserted into *mis_daily_files* at the beginning of each MIS Extract day for each data file to be delivered. Any process which either populates the staging table for one of these files or creates the Data Warehouse Interface file must update the relevant *mis_daily_files* row to indicate how far the processing has completed and how many rows were extracted/created.

6.1.4 Warehouse File Repository Switch

All files delivered to the data warehouse will be placed in a directory on a NFS partition which will reside on the data warehouse platform. The base directory name will be identified by concatenating MIS system parameters DWH-REPOSIT and DWH-SYSTEM. Within this, there will be a further sub-directory for each day to be delivered which will be named according to the date of delivery in the form 'YYYYMMDD'.

Should the communication link between the PAS/CMS platform and the data warehouse platform be lost, the data warehouse files must be delivered to a UNIX directory residing locally. It will be a function of the operations staff and support to determine whether the fault is due to a communications failure and to manually switch the mode of operation to local delivery. The local directory will be determined by concatenating MIS system parameters LOC-REPOSIT and LOC-SYSTEM.

A separate system parameter will determine whether delivery is being performed to the remote or the local platform. This may be switched using process MISC104 which is described in section 0 - 6.2.4 Fallback Delivery.

If delivery to the local platform is indicated, then the base directory indicated by LOC-REPOSIT & LOC-SYSTEM must exist. However, it is unlikely that the directory for the specific date of delivery will exist. This must be created by the first process which attempts to create a file within it. Also, if the LOC-SYSTEM part of the directory does not exist then the program should create it.

FUJ00117516

6.2 Extract Control Processes

6.2.1 New Extract Day

The MIS Extract day begins when the PAS/CMS active date rolls forward onto the next day. This process is performed by SUPC702 at the start of the POCL Core Day (08:00). The New Extract Day process should run shortly after SUPC702.

It is the responsibility of this process to set up the current days' staging tables and meta-data. The staging tables are a set of intermediate tables created each day in order to provide resilience in the event of data loss of the Data Warehouse flat files. These will be cleared-down after a pre-determined number of days as specified in system parameter MISPURGE.

The meta-data produced is a list of files to be delivered to the Data Warehouse today. This will act as an audit and ensure that what was expected to be delivered has actually been produced. In addition, a list of aggregates required by the Data Warehouse today will also be produced. Again, this ensures that all aggregates are delivered as expected.

6.2.1.1 Daily File Audit Creation

A row will be inserted into *mis_daily_files* for each row in *mis_data_files* with the date set to the current working day as held in *system_codes.code_date* where *code_type* = 'SYS' and *code_name* = 'TODAYS_DATE'. This set of information will provide an audit of data files delivered to the data warehouse. Rows with a frequency of 'W' & 'M' will only be created if the day of execution is at the end of the week or month correspondingly.

6.2.1.2 Daily Aggregate Audit Creation

A row will be inserted into *mis_daily_aggregates* for each row in *mis_aggregated_data* with the date set to the current working day as held in *system_codes.code_date* where *code_type* = 'SYS' and *code_name* = 'TODAYS_DATE'. This set of information will provide an audit of aggregated data attributes delivered to the data warehouse. . Rows with a frequency of 'W' & 'M' will only be created if the day of execution is at the end of the week or month correspondingly.

6.2.1.3 Staging Table Creation

For each row in *mis_data_files* which has attribute *staging_table* = TRUE, a new table will be created in the form:

MIS_DW_*xxx*_YYYYMMDD	Where *xxx* = *mis_data_files.data_name* and the table structure and storage clauses will be found in *mis_data_files.sql_script*.

Tables with a frequency of 'W' & 'M' will only be created if the day of execution is at the end of the week or month correspondingly.

A synonym will be created for each staging table created in the following form:

FUJ00117516

MIS_DW_*xxx* Where *xxx* = *mis_data_files.data_name*

6.2.2 Data Warehouse Delivery

The Data Warehouse requires that all files to be provided today are provided in total by 02:00 or not provided until the next day. The delivery mechanism must therefore execute at approximately 01:30 to ensure final delivery by 02:00. Final delivery will be deferred until the following day if this time is not met (see later). This final delivery time is held as a system parameter DELIVERY which indicates the number of minutes past midnight when the delivery is due. Should the requirement for deferred delivery be suspended, this parameter may be set to a high value.

The Delivery process will run after all other MIS Extract processes and the first phase will be to check that all direct extracts and the population of all staging tables has completed. This will be done by checking today's entries in *mis_daily_files*. If this is not so, the process will abend and the missing data must be provided by manual execution of the relevant extract process before re-running the delivery.

Aggregated data is only provided to the data warehouse once it is available and all available aggregates are provided in the same Data Warehouse file regardless of the source date of the aggregate. This is performed in three stages:

1. The rows in *mis_consolidated_aggregates* are summed by attribute where more than one row exists for the same attribute and those rows contain different process dates. The rows with the greatest process dates are ignored. The resultant data is placed, one row per attribute, in *mis_dw_aggregated_count* and deleted from *mis_consolidated_aggregates*.
2. The table *mis_dw_aggregated_count* will be locked and the entire contents inserted into the current staging table *mis_dw_act* after which *mis_dw_aggregated_count* is truncated and unlocked.
3. The information from *mis_dw_aggregated_count* needs to be enriched with two additional counts, Fndpd3 & Fndei3. These correspond to, and have the same counts, as Fndpd & Fndei respectively. If either Fndpd or Fndei now exist in mis_dw_act then they should be duplicated to create the corresponding rows.
4. The resulting rows in *mis_dw_act* will be used to update rows in the audit table *mis_daily_aggregates* setting the delivered flag and timestamp for each aggregate and process date.

All aggregates should be delivered in date order. If, as a result of the above processes, there are now rows in *mis_daily_aggregates* which have not yet been delivered and, for the same aggregate, a row of a greater process date has been delivered, then it will be assumed that the data for that aggregate is missing. In this case, an exception will be raised stating the aggregate name and process date and the offending row will be updated (*delivered* = 2) so that the error is not re-reported.

Regardless of whether errors were logged during the aggregating phase, the delivery process will extract all staging tables directly to the Data Warehouse interface files. Aggregated data will be enriched during this unload to append the *mis_aggregated_data.type* to each row.

FUJ00117516

With completion of extraction of the staging table data, all data required by the Data Warehouse is now in flat-file format in the current day directory. This needs to be described with the creation of a delivery control file which is described in detail in Reference [5]. The source of the information which is used to create this file is from *mis_dw_control* and *mis_daily_files*. Some clarification as to the contents of the control file follows:

Identifier	Description	Source
NUMFILES	The number of files to be delivered in this drop	A count of rows in *mis_daily_files* with a date equal to the current processing date
COMMENT	A comment	In the form 'CMS/PAS extract files for ORIGDATE'. The format of ORGIDATE is defined below
SOURCE	An entry for each file to be delivered giving the file identifier and count of rows in the file	An entry of *data_name* and *Qty Rows* for each row in *mis_daily_files* with a date equal to the current processing date
TARGET	The target system for the data	Set to DW
DELIVDATE	The date and time of delivery	This is the date/time of data availability to the data warehouse and is set to the system date at the point of creation of the control file
ORIGDATE	The current MIS Extract Date	The time will be set to 23:59:59.
NUMEXTRACTS	The number of other daily extracts made available today	A count of rows in *mis_dw_control* where *Delivered* is NULL.
OTHEREXTRACT DATES	A list of dates indicating what other extract are available for Data Warehouse upload	A colon-separated list of *Process Date* for each row in *mis_dw_control* where *Delivered* is NULL.
FILE	An entry for each file delivered today and the number of rows in each.	As with SOURCE above except that the filename is made up from the current MIS Extract Date plus *data_name*

A row will now be inserted in *mis_dw_control* with *Process Date* set to the Current MIS Extract Day and *Available* set to the current system date/time.

The data is marked as available to the data warehouse by the insertion of a lock file. If the current system time is more than system parameter DELIVERY minutes past midnight on the current Extract day then the delivery process will terminate without the creation of a lock file since the window of delivery has passed and the data should be made available on the next working day. If such time has not yet passed, a lock file

FUJ00117516

will be created in each directory relating to entries in *mis_dw_control* where *Delivered* is NULL - *Delivered* will be updated to the current system time for all such rows.

The delivery directory is part of an NFS partition which resides on the Data Warehouse platform. Creation of the days' directory and subsequent housekeeping is the responsibility of the data warehouse.

6.2.3 Purging

A purge process will execute daily after delivery of data to the data warehouse. This will truncate and drop all staging tables for which *mis_dw_control.delivered* is older than system parameter MISPURGE. Table *mis_dw_control* will be updated with the date/time of purge and all rows in *mis_daily_files* and *mis_daily_aggregates* for that day will be deleted.

It is possible for New Extract Day process MISC101 to abort due to errors such as tablespace exceeded. Since the New Extract Day creates new tables and synonyms, it is impossible for this process to rollback completely. In order for the database to be restored to its original state, the purge process may be invoked with an argument which will indicate purging of the current MIS Day. This will first check *mis_daily_files* to ensure that no data has yet been delivered and will then purge only the current day.

6.2.4 Fallback Delivery

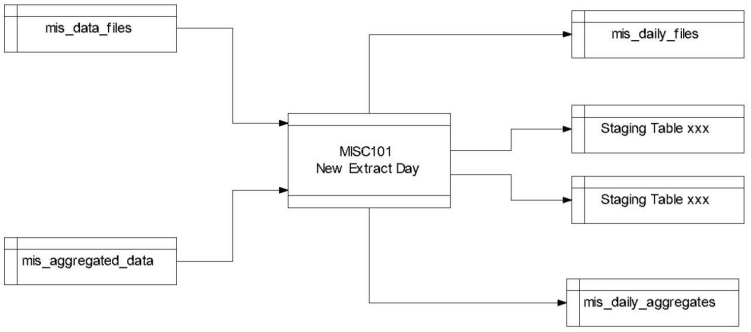
In the event of the link to the data warehouse platform failing, the MIS Extract must be capable of writing the delivery files to a directory on the local platform. This will be performed by switching the MIS system parameter MIS-PLATFORM to 'LOC'. This will indicate to the delivery processes that the system parameters indicating the delivery directory will be LOC-REPOSIT and LOC-SYSTEM.

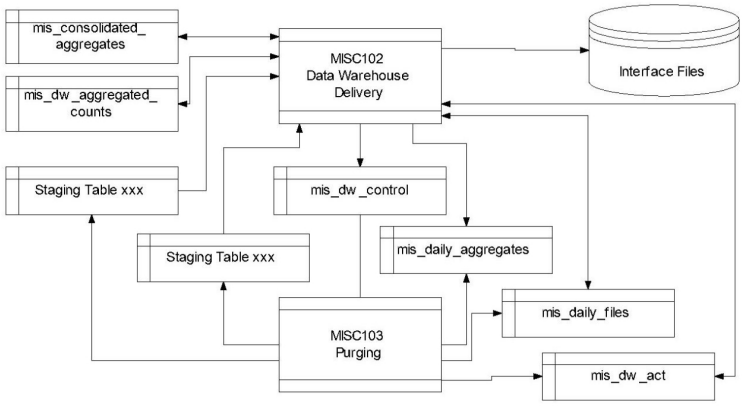
This process will also be capable of switching the parameter back to 'DWH' which will indicate the use of DWH-REPOSIT and DWH-SYSTEM which describe the remote pathname on the NFS partition.

FUJ00117516

6.2.5 Extract Control Data Flow Diagrams

The following diagrams show the flow of information between the extract control processes and the underlying data.





FUJ00117516

6.3 Direct Extracts

Detailing all MIS Extract processes which create the Data Warehouse interface files directly from the source PAS/CMS tables.

6.3.1 Help Desk Calls

The Help Desk is a 24 hour operation and the MIS Extract will unload call transaction detail on a daily basis to the interface file described in 0 - 7.5.1 Help Desk Calls. All data extracted will be removed after extraction of the information. To achieve this, a number of call tables will be used with a synonym pointing to the current table. The number of *calls* tables held and the current table will be held as MIS system parameters (NUMCALLS, CALLS). The extract will act in two phases:

1. Truncate the inactive table then drop the synonym from the current table and re-create it on the other table thus switching the current table. The table from which the synonym was dropped will now be inactive

A synonym, CALLS_ARC, will point to the previous days CALLS_n table. It will be the data referenced by this synonym which will be archived by module SUPC313 (generic archiver). Therefore, whenever the CALLS table synonym is moved on, the CALLS_ARC synonym must be moved on accordingly.

2. The MIS Extract will unload the (now) inactive table directly to the Data Warehouse Interface files.

The current table will be held as a system parameter which will be set to 1, 2, 3 etc. At the end of Phase 1 of the process, the system parameter will be updated to be the next number in the list in a rotational manner. At this point, MIS system parameter CALLDATE will be updated to reflect the current PAS/CMS Day. In this way, re-execution of the extract after abort would first check the date in CALLDATE re-switching the synonym only if that date was prior to the current date.

The helpdesk function should not be affected by the synonym switch since calls are written and committed on wrap-up in a single action. However, the procedure which performs this will catch 'Object Does Not Exist' error and retry if necessary.

This process should run as close as possible to 20:00 to give the Data Warehouse a consistent view of daily transactions but otherwise has no dependencies.

6.3.2 Reference Data

Reference data will be supplied to the Data Warehouse from the following PAS/CMS tables:

- Payee Roles
- Benefit Agencies
- Benefit Types
- Call Wrap Codes (From *wrap_enquiry_actions*)

FUJ00117516

- Benefit Offices
- BA Profiles

The extract will access these tables directly and write all rows to the interface files in the form described in section 0 - 7.5 Warehouse Interface Files.

6.3.3 On-Line Interface

Urgent transactions will be performed using the on-line interface described in [**3**]. The measurement of SLA's will be performed from within CAS On-line which will record the start and finish time of each transaction to an accuracy of one microsecond.

The MIS Extract will provide the Data Warehouse with all on-line transaction detail.

The source information will be provided in the form of flat files delivered from each Benefits Agency system to a directory defined in system parameter 'ONLINE' (File Type = 501) and will be of the form described in [**3**] Section 5.2. Each detail record/line within the Audit file will relate to a single transaction which will consist of a main transaction followed by zero, one or more transaction components. The areas of interest for the MIS extract are as follows:

Field Position	Name	Picture	Description
3-20	Start Time	9(18)	The time in microseconds since 31st Dec 1899 of when CAS takes control
21-38	Finish Time	9(18)	The time in microseconds since 31st Dec 1899 of when CAS relinquishes control
39-56	Sequential No	9(18)	Sequential transaction identifier
134-137	No of Business Functions	9(4)	A description of what constitutes a Business function can be found in [**3**].
164-168	Error Code	X(5)	The coding of any fatal error occur which may have occurred within the transaction
169-203	The following 2 attributes are repeated 5 times		
1-3	Scorecard Component	9(3)	Described in detail below. Zeros if not populated.
4-7	Component Count	9(4)	The count of the scorecard component. Zeros if not populated.

Of the repeated Scorecard components, the first component will be the main transaction which may only be one of the following:

Component	Description
280	Payment Authorisation
370	PAS Customer Detail
290	Stop Payment

FUJ00117516

301	Temporary Token Issue
302	Urgent Proxy Enabling

In addition to the five types of main component, there are an additional three types of component:

Component	Description
281	Payee Group within Authorised Payment
282	Token Group within Authorised Payment
380	CMS Customer Detail

The Data Warehouse requires transaction detail extracted from the Audit files in the form described in 0 7.5.2 On-Line Main Business Transaction and 0 -7.5.3 On-Line Transaction Detail. All of this information is directly available from the Audit files and will be written directly to the MIS Extract Flat File Area.

The On-line day is between 08:00 and 20:00 and it is assumed that the Audit file will be created outside of these hours. The originating date of the flat files to be delivered to the Data Warehouse will be the Creation Date held within the Audit Header Record *Less* 8 hours.

This means that an Audit File created between 20:00 on day A and 08:00 on Day A+1 will be assumed to be the product of all transactions on Day A. Should problems occur on the VME mainframe which cause the creation of the Audit file to be delayed beyond 08:00 on Day A+1, then the MIS Extract will not be able to differentiate between Day A transactions and Day A+1 transactions. In this instance, Day A and Day A+1 transactions will be delivered together in Day A+1.

6.3.4 Post Office Temporary Closure

Post Office information is maintained by PAS/CMS Reference data maintenance process CMSC112 and the Help desk. These processes write out an audit of changes to *po_events* and it is this data which will indicate temporary closure and re-opening times. Since this data is modified by on-line maintenance programs and the help desk, Post Office change information may be entered at any time during the day (even off-line). The data extracted should therefore be for *yesterdays* events and will recognise all events with *status* = [T]emporary closure, [O]pen, and [E]mergency closure. This will be scheduled to run shortly after the start of the MIS Extract day, has no other dependencies and writes directly to the interface file described in 0 - 7.5.15 PO Temporary Closure.

Re-execution of this process after abend is straightforward.

6.3.5 Changes of NPO Not Reversed

A new PAS process will be responsible for recognising individual instances of change of Nominated P.O. not reversed within 6 weeks, where there is encashment of means-tested or specified non-means tested benefit whose date of availability is after

FUJ00117516

change of NPO and there is no notification of change of address received within 6 weeks. Those instances which are recognised by the PAS process will be written to *mis_dw_unrev_npo*.

The 24hr Help Desk will one of the PAS/CMS processes which are partly responsible for recording changes of NPO and changes of address. These changes may be applied at the 11th hour (late in the day after 5 weeks and 6 days of changing NPO), and cause a record not to be reported. The new PAS process must therefore run after midnight and is assumed to run during the overnight batch schedule.

The Changes of NPO Extract will therefore run during the next on-line day run shortly after the start of the MIS Extract day and has no other dependencies. All rows from *mis_dw_unrev_npo* with a creation date of the previous day will be extracted and written directly to the warehouse interface file described in 0 - 7.5.18 Changes of NPO.

Once successful, all rows with a creation date older than the number of days in system parameter NPOPURGE will be deleted. This ensures recoverability in the event of system failure.

6.3.6 Temporary Tokens Issued/Impounded/Unused

A count of temporary tokens Issued/Impounded/Unused per DSS Issuing Office is required on a daily basis. This process will identify *temporary_token_events* for the day of the Extract and will categorise the following Event Codes:

Event Code	Description	Count As
07	Token Assigned	Token Assigned
10	Token Expired	Token Unused (If the associated 'used' flag on the Temporary Token has not been set)
05	Token Retained	Impounded
12	Token Defaced/Altered	Impounded
13	Counterfeit Token Suspected	Impounded
14	Token Not Known to System	Impounded
15	Customer Left PO During Transaction	Impounded
16	Suspicious Proof of Identity	Impounded
17	Poor Signature Match	Impounded
18	Other Impound Reason	Impounded

Not used means 'Expired and Not Used'. PAS/CMS process PMSC314 - Temporary Token Marking will indicate on the temporary token whether the token has been used to encash. The Extract will identify all temporary tokens which have expired on the date of execution which have not been used.

FUJ00117516

Temporary Tokens are populated well in advance of when the Extract requires the information, after which a number of events will be performed finally resulting in the Temporary Token being STOPped by an Expiry, Impound or other event.

Temporary Tokens are purged (SUPC310) by book after all Tokens have been assigned or stopped and *n* days (as indicated by PAS/CMS system parameter *temp_token_archive*) has passed, *and* where all Tokens have their archive flag set. The archive flag is set when the token is expired or stopped.

The expiry period of Tokens is determined by another PAS/CMS system parameter. If this is the same or greater than *temp_token_archive* then it is possible that the Token could be purged before the MIS Extract has the opportunity to perform the Counts. *It is therefore essential that temp_token_archive is set to at least 3 days greater than the token expiry period.*

Based on this assumption, the Temporary Token Extract can be performed directly to the Data Warehouse flat file interface (0 - 7.5.21 Temporary Tokens Issued) and be recoverable in the event of data loss by re-running for the required day.

Since *temporary_token_events* are populated by both the 24hr Helpdesk and a number of PAS/CMS overnight batch processes, the extract will operate one day in arrears to ensure that all events for the date of extract have been processed.

6.3.7 Notification of PUNS Not Received

A function of the PAS/CMS Helpdesk will be to log all those customers who report that they have not yet received a PUN and yet have already collected their Card. This event-driven information will be logged into *mis_repudiated_puns* and extracted daily one day in arrears to the interface file described in 0 - 7.5.16 PUNS Not Received.

The Extract will be responsible for removing the rows in *mis_repudiated_puns* once they have aged beyond system parameter PUNPURGE this provides resilience in the event of extracted data loss.

6.3.8 Card Production

In order to measure SLA's for card production and delivery, the Data Warehouse requires card and batch information delivered on a daily basis. As each batch is made available for collection by the customer, each card delivered in the batch can be analysed for conformance to the SLA by comparing the original order date with the batch reconciliation time (when the cards associated with the batch are assumed to be available for collection). Other batch events are also delivered to the Data Warehouse so that individual stages in the production and delivery of batches may be measured. Only batches with a batch type of 'C' will be evaluated for extraction and only the following batch events are delivered:

03 - Batch Received @ Post Office
07 - Batch Ordered
10 - Batch Reconciled
14 - Delivered to private box number
15 - Delivered to destination address
16 - Collected by recipient from callers office

FUJ00117516

25 - Despatched from De-La-Rue

The initial instigation for extraction of batch information is the receipt of a despatch batch event 10 - Batch Reconciled. This can be recognised by despatch batches with a status of 'BKD' and a NULL extract date (or BKD and an extract date = today in the case of recovery). For each of these batches, the batch details and all associated batch events will be extracted along with summarised despatch batch card information. For each batch extracted, the table *despatch_batches* will be updated setting *extract_date* to the current MIS Extract Date and *last_timestamp_extracted* to the current system date.

For batches which have been previously extracted, any batch events which occur after the extraction date must also be extracted. This will be performed by identifying all todays events (*creation_timestamp*) for batches where status = 'BKD' and where the despatch batch extract date is not NULL (and the extract date < Today in the case of recovery). For each batch processed, the table *despatch_batches* will be updated setting *last_timestamp_extracted* to the current system date.

In order to ensure that all despatch batch events have been processed prior to extraction, the despatch batch extraction process will execute one day in arrears and may run at any time after the Start of MIS Day.

This extract delivers data directly to the warehouse interface files described in sections 0 - 7.5.11 Despatch Batches, 0 - 7.5.12 Despatch Batch Cards and 0 - 7.5.13 Despatch Batch Events.

6.3.9 Random Encashment Record

A single row is inserted daily into *mis_enc_sample_details* containing a combination of the encashment record, encashed payment and payments encashed trailer as sent back to CAPS and as detailed in [1] for a randomly selected encashments record. This row also contains additional information including the Benefit Type and some details on the criteria and result of the random method. The process which creates this row is PMSC310 - Produce CAPS File of Encashments which may run at any time during the batch overnight schedule or even the day following encashment.

The MIS Extract will therefore run reasonably late in the MIS schedule in an attempt to extract the row created for the previous days' transactions. All rows which have a NULL extract date will be extracted directly to the Warehouse interface file (see 0 - 7.5.22 Random Selection of Encashments) and the extract date will be updated with the current MIS Day. If no row is found, an empty extract file will be produced.

Re-run after data loss will extract all rows in *mis_enc_sample_details* where the extract date is the same as the re-run date. Rows in *mis_enc_sample_details* will be purged after they have reached an age greater than the number of days specified in system parameter MIS-ENCPURGE

6.3.10 Customers Infringing Change of NPO

Customers infringing change of NPO will not form part of the PAS/CMS MIS Data Extract since all attempts made will be transmitted to the Data Warehouse using Riposte messages.

FUJ00117516

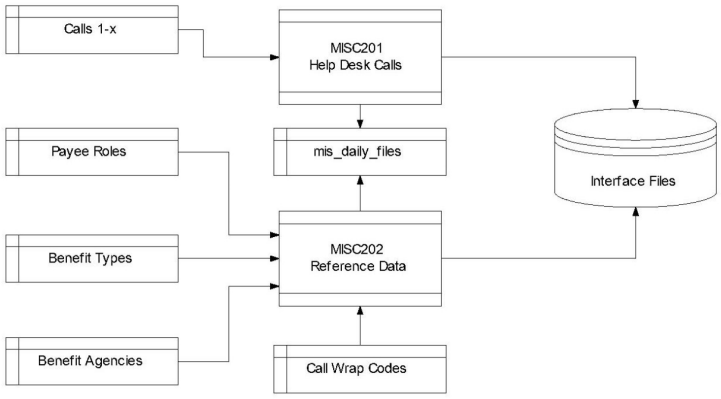
6.3.11 Attempts to Encash Breaking RPOI

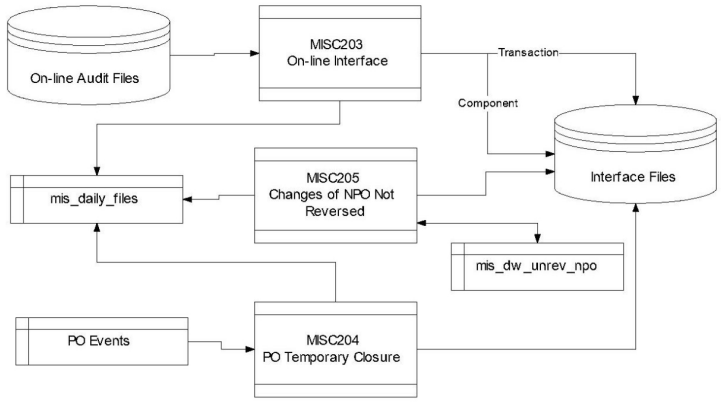
Attempts to encash breaking RPOI will not form part of the PAS/CMS MIS Data Extract since all attempts made will be transmitted to the Data Warehouse using Riposte messages.

FUJ00117516

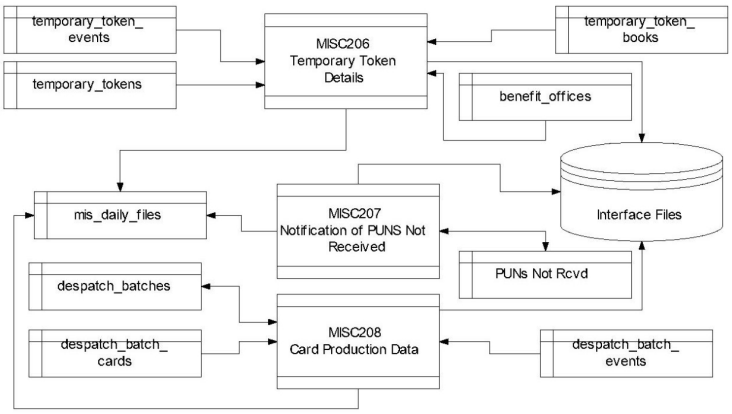
6.3.12 Direct Extract Data Flow Diagrams

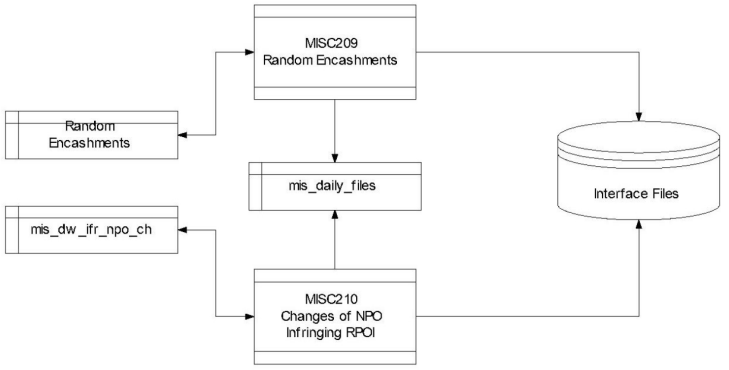
The following diagrams show the flow of information between the direct extract processes and the underlying data.





FUJ00117516





FUJ00117516

6.4 Two-Phase Extracts

Detailing all MIS Extract processes which create the Data Warehouse interface files via a set of interim staging tables in order to provide resilience or performance enhancements.

6.4.1 Ongoing Active Cards

The number of active cards per Post Office is required at the end of each week. This is essentially a snapshot of the total number of cards active at a specific point in time and can be determined by matching the time of activation and stop events against cards and card status. This strategy would consume too much resource and instead, the Extract will simply count the number of cards with an active status at the time that the process is run.

In order to get a consistent count on a weekly basis, the PAS/CMS processes which activate and deactivate cards must be considered and the count of active cards should be performed after the current days' card events have been processed.

Process	Description	Comments
CMSC109	Cardholders Not of Interest	Overnight batch process
HLPF202	Helpdesk	Card stops issued from 24 hr helpdesk
CMSC301	Card Collection and Impound Events	Received and processed from the Post Offices during the POCL Core Day

Since the Helpdesk is a 24hr activity, it is impossible to get a truly accurate figure, but the number of card stop calls outside of the POCL Core day is not assumed to be significant.

The extract should therefore run after CMSC109 which should have completed by about 01:00 to allow the active card data to be extracted and finally delivered to the Data Warehouse by 02:00. The Extract will place the results into the staging table *mis_dw_cda*.

6.4.2 Cards Issued Per Post Office

A PAS/CMS process, CMSC201 - Produce Cards and PUNS, is responsible for populating the MIS Extract table *mis_ordered_cards*. This process runs twice in the overnight batch schedule; once for urgent card orders and secondly for non-urgent orders. The MIS Extract process which processes this information will not guarantee to pass the current days' information to the Data Warehouse by 02:00. This process will therefore execute early in the MIS Extract Day and process the previous days' Card Issues prior to truncating *mis_ordered_cards* ready for re-population later that day. Issued Card information will therefore arrive in the data warehouse one day late.

In order to retain resilience, the extracted data will be grouped by Post Office, Card Design Type and Issue Reason and placed into the staging table *mis_dw_cdi*.

FUJ00117516

6.4.3 Cards Deactivated Per Post Office

A PAS/CMS process, CMSC115 - Card Event Partitioning, is responsible for populating the MIS Extract table *mis_inactivated_cards* on a daily basis. This extract process will therefore execute after completion of CMSC115 and will process the current Card Deactivations prior to truncating *mis_inactivated_cards* ready for re-population. The process will run on a weekly basis and will process all rows in *mis_inactivated_cards*.

In order to retain resilience, the extracted data will be grouped by Post Office and Deactivation Reason and placed into the staging table *mis_dw_cdd.*

6.4.4 Card Events

The data warehouse requires certain individual card events to be supplied. These events are all supplied directly into the table *mis_dw_card_events* by the PAS/CMS process, CMSC115 - Card Event Partitioning which executes during the POCL Core Day. The event types are described in the data definitions.

The MIS Extract process will be scheduled to run immediately after CMSC115 and take the contents of *mis_dw_card_events* in their entirety and insert them into the staging table *mis_dw_cr1* prior to truncating the source table. During the extract, the *linked_office* and *NINO* attribute will be added to each card event by reference to the Post Office reference data *post_offices* and the *cards* table. Also, the Nominated post office *cardholders.po_code* will also be added.

6.4.5 Benefit Non-Encashment

This process extracts details of Payments not encashed within 4 weeks of due date - Only for means-tested benefits - And also - Details of Payment not encashed within 6 weeks of due date - For means-tested benefits and Specific non-means tested benefits.

The source of this information is the PAS/CMS *payments* table in association with the *benefit_types* table which provides a means tested indicator and a specified non-means tested indicator. The *payments* table is updated with encashment information by the PAS/CMS Process SUPC301 - Merge & Purge Payments which must execute before performing the extract. Since SUPC301 executes late in the overnight batch schedule, the extract will use the previous days' date to perform the date comparison and will therefore extract individual details of rows within *payments* which match the following criteria:

- Yesterday minus *payments.earliest_encash_date* = 29 and (Means Tested)
- Yesterday minus *payments.earliest_encash_date* = 43 and (Means Tested or Specified Non-Means Tested)

The individual rows located will be inserted into the staging table *mis_dw_cr2*

6.4.6 Encashments Made After Payment Stop Received

The PAS/CMS table *encashed_pay_excptns* will hold the payment details which were encashed after a payment stop had been received. The payment and encashment ID's can be extracted from those rows which have an encashed-after-stop status (*excptn_code* = 33) by interrogating the *excptn_detail* and extracting them as a sub-

FUJ00117516

string. The encashments table *tms_rx_encashments* and *payment_events* (via *tms_rx_enc_payments*) will be accessed to yield the additional information required by the Data Warehouse. This report will run daily against the timestamp in *encashed_pay_excptns*.

The stop request will not exist in *payment_stops* since the payment would have already have been received prior to the stop request. Instead, a provisional payment event will have been raised in *payment_events* with status 'STP'. The stop date/time must be retrieved from this table using *payment_id* and *status_code*.

The *encashed_pay_excptns* table is populated by the PAS/CMS process Validate Encashed Payments - PMSC302 which executes at a late stage in the overnight batch. The Extract will therefore not be able to guarantee the delivery of Todays exceptions to the Data Warehouse and will run one day in arrears by selecting all *encashed_pay_excptns* with a timestamp of yesterday.

This complicates issues slightly since *tms_rx_encashments* and *tms_rx_enc_payments* are synonyms which are rotated on a daily basis to point to the next table in the sequence of *encashments_1-3* and *encashed_payments_1-3* giving up to 3 days of encashment information at any time (refer to [4] for a full description). The synonyms are dropped and re-created against the next-days table (suitably truncated) by the PAS/CMS process Encashments Housekeeping - SUPC305 after all overnight processing has completed.

This means that when the Extract runs the next day, the synonyms will not be pointing to the correct tables. Yesterdays table name can be extrapolated by interrogating the PAS/CMS *system_parameters* table for the following:

Parameter Type	Parameter Name	Description
PMS	MAX_ENCASHMNT_ HISTORY	The number of *encashments_x* tables held at any one time
PMS	ENCASHED_PAYME NTS	The current *encashments_x* table. Stored as the number x.

Subtract 1 from ENCASHED_PAYMENTS and set to MAX_ENCASHMNT_ HISTORY if the result is zero, this will give the number of the previous days' *encashment* and *encashed_payments* tables.

The extract may therefore run soon after the start of the MIS Extract Day and will extract information to *mis_dw_cr6*. The data definitions detail the source of each attribute.

6.4.7 Customers With Many Changes of NPO

This extract will list all customers who have registered *x* or more changes of NPO in *y* months with *x* and *y* defined by system parameters NPOTIMES and NPOMNTHS respectively.

FUJ00117516

The delivery of data to the Data Warehouse will run on the last day of each month and it would therefore be possible for it to miss certain instances that would otherwise be counted. For example, assuming 6 changes measured in 12 months:

Change of NPO: 10/01/98, 15/02/98, 20/05/98, 25/09/98, 30/11/98, 05/01/99

The above shows that 6 changes were made within a 12 month period yet running the report on either the 31st December 98 or 31st January 99 would not recognise 6 changes in the previous 12 Months.

The extract will therefore run on a daily basis and access *nominated_post_offices* evaluating *timestamp* against each NINO to produce the necessary counts into *mis_npo_changes*. On the last day of the month, and following the daily extract procedure, the rows within *mis_npo_changes* will be consolidated into the staging table *mis_dw_cr5* ensuring that only one row exists per NINO and that this row is the one from *mis_npo_changes* with the greatest *No of Times* recorded. The table *mis_npo_changes* will then be truncated.

The following aggregated count will always be placed in *mis_dw_aggregated_count* on the last day of each month:

Description	Aggregate Name
Number of months for which the counts were accumulated (NPOMNTHS)	NONPOMO

6.4.8 New Cardholders

This extract will list all new cardholders registered in a given period. This period is assumed to be one month.

The extract is based on table mis_new_cardholders. The invalid_address_marker (known as DLO indicator to the data warehouse) and stop_event on mis_new_cardholders could be incorrect. Therefore in order to provide coherent data to the data warehouse, some data transformations are performed as the data is extracted from mis_new_cardholders. Each scenario covered, with the associated transformation, is given below:

1. Cardholder is not DLO but still does not have a row in the CARDS table.

   Transformation: Set DLO indicator to 'N'

2. Cardholder is DLO but has a valid card

   Transformation: Set DLO indicator to NULL.

FUJ00117516

3. Cardholder is DLO but has a card stop event recorded (has been issued with a card which has been stopped with no re-order)

   Transformation: Set DLO indicator to NULL.

4. Cardholder is not DLO, has a stop event recorded, but has had another card ordered which is valid (not stopped).

   Transformation: Set stop event to NULL.

The extract should be run after the card order processes (urgent and non-urgent), namely CMSC201, and the card event population process (CMSC115) in the overnight schedule. The extracted data is loaded into the staging table mis_dw_cqx.

6.4.9 Temporary Tokens out of Stock

This extract runs daily and locates DSS offices that have run out of stock of Temporary Tokens today, or have until today, been out of stock.

The data warehouse only wishes to be told when:

i) An office goes out of stock

and ii) An office goes back into a replenished state.

The data warehouse does not wish to be informed on each day that an office is out of stock, merely at the start and end of having no stock.

In order to identify when a DSS office has had it's stock replenished, a store is kept of the offices that are currently out of stock.

Thus, on each day the offices that are out of stock are identified and treated as follows: If an office has not been out of stock on a previous day, then it is included in the extract with event type 'O' (for Out of stock). This office is then added to the store of 'offices out of stock'. If an office has been previously out of stock then it is ignored today.

If an office that has been previously out of stock is not identified today as being out of stock (ie. no longer out of stock), then it is included in the extract with event type 'R' (for Replenished). This office is then removed from the store of 'offices out of stock'.

To achieve this functionality, two intermediate tables are used: mis_tt_stock_out_today and mis_tt_stock_out_prior.

Table mis_tt_stock_today will be truncated and then populated with the offices that are out of stock today. Following this, rows in mis_tt_stock_out_today which do not exist in mis_tt_stock_out_prior are reported to the data warehouse as event_type = 'O'. These rows are then inserted into mis_tt_stock_out_prior.

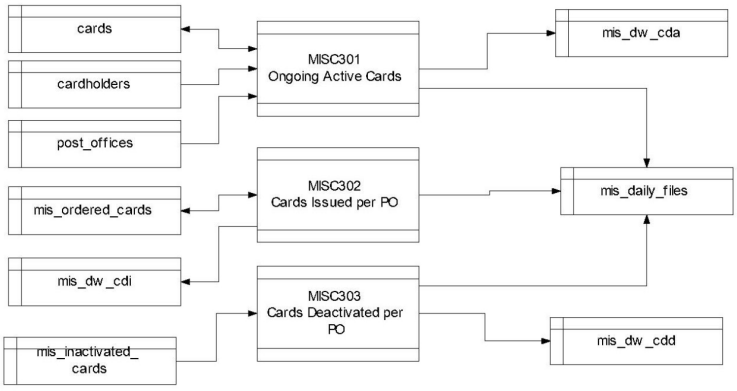
FUJ00117516

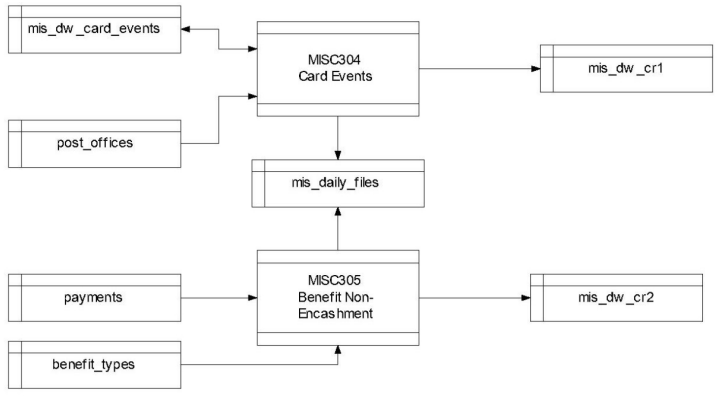
Rows in mis_tt_stock_out_prior which do not exist in mis_tt_stock_out_today are reported to the data warehouse as event_type = 'R'. These rows are then deleted from mis_tt_stock_out_prior.

This extract should be run after the PAS/CMS Temporary Token re-order process (CMSC401). The extracted data is put into staging table mis_dw_bax.

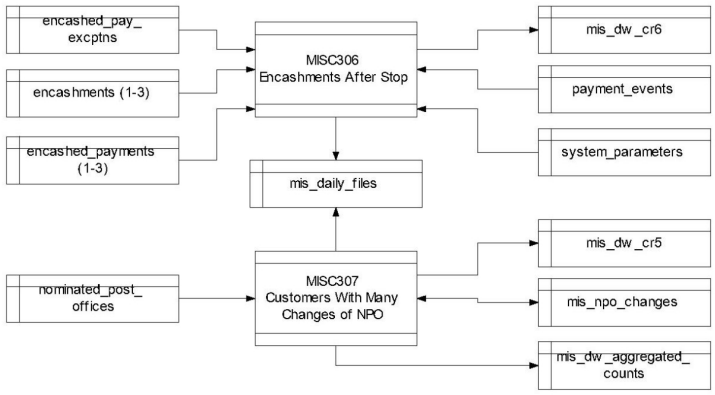
6.4.10 Two-Phase Extract Data Flow Diagrams

The following diagrams show the flow of information between the two-phase extract processes and the underlying data.





FUJ00117516



6.5

FUJ00117516

Aggregated Data

Aggregated data is produced by batch and event driven processes. This data may be produced either overnight or during the POCL Core Day by TMS Agents, PAS/CMS processes and MIS Extracts. The data must therefore be accumulated in a common area (*mis_dw_aggregated_count)* prior to eventual extraction into the current staging table (*mis_dw_act*) and its' eventual unload to the Data Warehouse Interface Files.

In addition, the Batch Acceptance/Rejection process (Section 0) produces more than one aggregated count per attribute per day. These counts have to be held-back until all the counts for an individual day have been obtained. The counts for day A will not be inserted into *mis_consolidated_aggregates* until the first count for day A+1 has been obtained after which the counts for Day A will be aggregated to *mis_dw_aggregated_count*.

6.5.1 Contingency Payments

If the link between PAS/CMS and the Benefits Agency fails, Pathway will issue payment authorisations based on the last payment date and frequency as maintained in *payment_mandates*.

The data warehouse has all of the encashment information from TPS. However, trawling through this information on a daily basis would be arduous. PAS/CMS will therefore provide a count in *mis_dw_aggregated_count* which will indicate whether there were *any* encashments made under contingency arrangements today.

PAS/CMS process PMSC310 'Produce CAPS File of Encashments' will write the count (CPAENTDY) directly to *mis_dw_aggregated_count* on a daily basis. The count will be aggregated by the date of encashment rather than the date of processing of PMSC310 which may be during the following day.

6.5.2 Summary Data

The following summary data will be provided on a monthly basis:

Aggregate Name	Description
NOBAC	Total Active BA Beneficiaries
NOBAA	Total Active BA Agents
NOSSAC	Total Active SSA (NI) Beneficiaries
NOSSAA	Total Active SSA (NI) Agents
NOWPAC	Total Active WPA Beneficiaries
NOWPAA	Total Active WPA Agents

A full definition of the requirements of the Data Warehouse may be found in [5]. The extract will access PAS/CMS table *payment_mandates* for all payments with an earliest encashment date within the current month of reporting and categorise each row by beneficiary and payee.

FUJ00117516

The payment mandates table is updated with new and updated payment details late in the overnight batch schedule. The summary data will therefore be provided on the 1st of each month and will provide information which relates to the state of play at the end of the previous day. This process may run shortly after the Start of the MIS Day.

6.5.3 Batch Timeliness CAPS to PAS

This section deals with the timeliness of distribution and availability of information passed from CAPS to PAS/CMS. The following types of information will be analysed:

1. Next-day Payment Authorisations
2. Next-day Stop Requests
3. Next-day Personal Detail Notifications
4. Next-day End of Interest Notifications
5. Regular Payment Authorisations

Of the above, the timeliness of distribution of items 1, 2 and 5 will be measured from CAPS to the point of delivery to the correspondence servers. Items 3 and 4 will be measured to the point where the data becomes available to the rest of the PAS/CMS system since the data received is not forwarded-on to the Post Offices.

For all types of transaction, the Data Warehouse requires counts of the Total number of transactions, the number of transactions not available at the destination by 08:00 and the number of transactions not available by 11:00 on the next working day. Measurement of conformance is only performed if the transactions are received from CAPS prior to 20:00 on the previous day. These times are held as system parameters (SLA-FCUTOFF, SLA-BATCH & SLA-BATCH2)

Payment Authorisations will only be measured for timeliness conformance if the Earliest Encashment date is equal to or less than the delivery date. Any Payment Authorisations which are received after Earliest Encashment date will not be measured for conformance to the SLA.

6.5.3.1 Payment Authorisations - Next Day & Regular

Payment Authorisations will contain a *payment_type* indicator which will be set to one of the following by a PAS process prior to passing the information to the TMS Agent via *tms_tx_payments*:

Code	Type	Description
N	Next Day	Payment received from CAPS prior to 20:00 and receipt date is the day prior to the earliest encashment date.
R	Regular	Payment received from CAPS more than one day prior to the earliest encashment date.
O	Other	Payment received from CAPS after 20:00 and receipt date is the day prior to the earliest encashment date or the receipt date is the same day or later than the earliest encashment date. Payment is a Re-issue or On-line payment.

FUJ00117516

The aggregated counts which record adherence to the SLA's will be sourced directly from the TMS Agent and the PAS Enrichment Process as well as from a MIS Extract process. The responsibilities of these processes is as follows:

Source	Description	Aggregate Name
PAS	Count of any Payments which are not passed to the TMS agent but are held back because no instrument of payment (Card or Temporary Token) is available	PYNOTPSD
TMS	Count of Next Day Payments not processed before the Start of POCL Core day (08:00) on earliest encashment date	Fndap
	Count of Next Day Payments not processed within 3 hours of Start of POCL Core day (11:00) on earliest encashment date. This figure will include the count from Fndap.	Fndap3
	Count of Regular Payments not processed before the Start of POCL Core day (08:00) on earliest encashment date	Tndpa
	Count of Regular Payments not processed within 3 hours of Start of POCL Core day (11:00) on earliest encashment date. . This figure will include the count from Tndpa.	Fndt3R
Extract	Total Next Day Payments passed to TMS	Tndap
	Total Regular Payments passed to TMS	TndnaA
	Total Other Payments passed to TMS	NOOTHER

All of the counts are based on the number of payment authorisations either passed to (or held-back from) the TMS Agent via *tms_tx_payments*. The process which produces this table, PMSC215 Payment Enrichment, also produces the PAS aggregated count and executes within the main stream of the overnight batch.

The other counts are dependent on *tms_tx_payments* being created and complete and cannot therefore start until PMSC215 has completed successfully. The MIS Extract process could run immediately after PMSC215 however the TMS Agent will not execute until the end of the overnight batch run and may continue well on into the next day (this being the reason for measurement). The aggregated counts from each of these processes must bear the same date which must be the Initial Receipt Date from CAPS. *It is therefore assumed that no files of payments will be processed between midnight and the time of execution of PMSC215.*

Counts PYNOTPSD, Tndap, TndnaA and NOOTHER will be written directly to the *mis_dw_aggregated_count* Table. However, since many TMS agents will be running simultaneously, there will be duplicate counts for Fndap, Fndap3, Tndpa and Fndt3R. These counts will be written to *mis_consolidated_aggregates* and will be subsequently aggregated together by MISC102 once the following day's aggregates start appearing.

6.5.3.2 Next Day Stop Requests

FUJ00117516

Payment Stop Requests will be passed to the TMS Agent via the *tms_tx_payment_stops* interface table. This will be populated with the date/time of the Stop Request as it passed across the CAPS boundary. When the TMS agent processes this table, it will insert a corresponding row into for each row processed in *tms_tx_payment_stops*. The TMS Agent enriches the payment stop information with a timestamp as the Payment Stop is passed to the Correspondence Servers.

In the situation when PMSC403 attempts to stop a payment where all the authorisation have been recall confirmed, PMSC403 will populate the *tms_rx_payment_stops* table directly and will populate the following columns in the RX table as follows :

initial_receipt_date : NULL
process_timestamp : NULL
event_source : PMSRCL

The MIS Extract will be scheduled to run after the TMS Payment Stop Interactive Loader has completed the transfer of the Payment Stops and will count the following from the *tms_rx_payment_stops* interface table.

Description	Aggregate Name
Total number of Stop Requests with actioned indicator set to NULL and Event Source = 'B'. Plus Total number of stop requests where event source = PMSRCL and receipt & process datetime columns are NULL	Tndsr
Total number of Stop Requests with actioned indicator set to NULL, Event Source = 'B' and Receipt Date less than 20:00 on the previous day and Actioned Date later than Start of POCL Core day (08:00)	Fndsr
Total number of Stop Requests with actioned indicator set to NULL, Event Source = 'B' and Receipt Date less than 20:00 on the previous day and Actioned Date later than 3 hours beyond Start of POCL Core day (11:00)	Fndsr3

In addition, the main Payment Stops process (PMSC403) will write the following count:

Description	Aggregate Name
Total number of Stop Requests not passed to TMS because the Payment was already stopped	PMSHELD

All counts will be written directly to the *mis_dw_aggregated_count* Table.

6.5.3.3 Personal Details and End of Interest Notifications

Both of these types of transaction are measured from CAPS to the point at which they are made available to the rest of the PAS/CMS system. Since the processing is

FUJ00117516

performed and committed by PAS, it will be those processes which produce the following aggregated counts:

Transaction Type	Total Processed Today	Total Processed where receipt date < Today
Personal Details Notification	Tndpd	Fndpd
End of Interest Notification	Tndei	Fndei

These counts will be written directly to the *mis_dw_aggregated_count* Table by processes PMSC103 (Personal Details) and PMSC108 (End of Interest).

6.5.4 Batch Acceptance/Rejection

This process measures the timeliness of acceptance/rejection of *data* files which are passed across the CAPS to PAS/CMS boundary. The process of *data* file delivery and acceptance/rejection is described in [**2**]. In summary, files of data are passed to PAS/CMS from CAPS and will contain records of one or other of the following types:[1]

Commented [PJ1]:

Commented [PJ2]:

File Type	Description
001 101	Payment Authorisation Next Day Payment Authorisation
002 011	Payment Stop Request Duplicate Payment Stop Request
012	PAS Personal Details Notification
014	CMS Personal Details Notification
013 , 015	End of Interest Notification corresponding to PAS Personal Details Notification and CMS Personal Details Notification respectively

Each *data* file undergoes validation and acceptance in a number of stages:

Receipt Type	Description
Initial	When the file is initially received
Acknowledgement	When the checksum and the format of the file structure and content have been checked as thoroughly as possible without the use of reference data
Acceptance	When all transactions have been fully validated

[1] Listed are only those file types which are of interest to the extract process

FUJ00117516

It is the timeliness of this acceptance and acknowledgement which is the subject of this process. In brief, for any *data* file delivered to us before 20:00, PAS/CMS must acknowledge the file by 22:00 on the same day and Accept/Reject the Transactions by 02:00 on the following morning. For example: If a *data* file is delivered to us after 20:00 on Day A, then PAS/CMS must acknowledge the file by 22:00 on Day A+1 and Accept/Reject the Transactions by 02:00 Day A+2. See system parameters SLA-FCUTOFF, SLA-FACCEPT & SLA-TACCEPT which define these times.

The following aggregated data must be supplied:

File Type	Total Files	Files Not Acknowledged By 22:00	Total Records	Records Not Accepted By 02:00
001 & 101	Tpaf	Fpaf	Tpan	Fpan
002 & 011	Tpsf	Fpsf	Tpsr	Fpsr
012	Tpdf	Fpdf	Tpdn	Fpdn
014	Ttacf	Fttf	Ttac	Ftt
013 & 015	Teoff	Feoff	Teoin	Feoin

File type 101 above refers to *data* files of Next-Day Payment Authorisations. However, it cannot be guaranteed that next day payment authorisations are not also supplied in files of type 001. Since these two transaction types are mixed in the same file type, it is impossible to measure the transaction acceptance times separately. Because of this, the aggregated counts for next-day payment authorisations will be added to the counts of regular payment authorisations and supplied to the Data Warehouse as a single count.

In order to verify that all *data* files that have been sent have also been received, an audit of the transmitted files is accumulated in Inward and Outward Control Logs in CAS (VME). Corresponding Inward and Outward Control Matching Files (ICMF/OCMF Files) are produced by a separate housekeeping process as and when it is scheduled. The format of these ICMF & OCMF files is described in [**1**]. One record exists in these *control* files for each *data* file transmitted across the CAPS to PAS/CMS boundary and will contain Date/Time stamps which will enable the extract to determine the initial delivery time, the acknowledgement/acceptance times and the number of records processed within each *data* file. Records will only be recorded in the *control* file after all processing of the *data* file has completed.

The ICMF and OCMF control files are delivered daily from the VME mainframes to a pre-defined directory (File Types = 021 & 022). The directory will be identified to the extract process by means of an argument passed at runtime.

Each line/record of the ICMF/OCMF files contains a number of fields. The extract is only interested in the following:

Field Position	Name	Picture	Description

FUJ00117516

8-10	File Type	999	The File Type
62-75	Initial Status Change Time	9(14)	The date/time that the *data* file was first passed across the interface. The data file delivery time.
77-90	Acknowledge Status Change Time	9(14)	The date/time when the data file was acknowledged
98-111	Status Change Time	9(14)	The date/time when acceptance of the data file took place
113-120	Created Transaction Count	9(8)	The total number of records/ transactions in the *data* file

The date/time fields are in the form YYYYMMDDHHMISS.

The ICMF & OCMF files are delivered to the Sequent on a daily basis from each of the Benefits Agency feeder systems. These *control* files will be delivered at some time after midnight but the actual time cannot be guaranteed since the schedules on the Benefits Agency systems are independent of Maestro and each other . In addition, since a *data* file may be received prior to 20:00 but not fully processed prior to the delivery of the *control* files, the information relating to that *data* file will not be present in the daily *control* file until the following day. It is therefore likely that in order to aggregate totals for *data* files delivered on day A, the extract will need to process *control* files for both day A and day A+1.

The MIS Extract must also consider the possibility of the FTF link between the Benefits Agency systems and the Sequent being inoperable for a period. In this case, not only would the *control* files be delivered late, but statistics may not be available for any one day since no *data* files were transmitted in that day.

In all cases, the MIS Extract must only deliver complete days-worth of aggregated information to the Data Warehouse and must never deliver any one day without delivering the preceding day aggregates even when zero.

In order to achieve this, aggregated information from the *control* files available at the time of execution of this process will be extracted and placed into a consolidation table *mis_consolidated_aggregates*.

A separate process will examine the consolidation table and aggregate daily information into *mis_dw_aggregated_count* from that only after the following days' statistics have appeared.

For Example:

Attribute Name	Originating Date	Count
Tpan	10/11/1997	4000000
Tpan	10/11/1997	2000000

FUJ00117516

Tpan	11/11/1997	3800000

The second entry for Tpan is the residue of information for the 10th which was received in the *control* file on the 11th. Once statistics for the 11th have appeared in the *mis_consolidated_aggregates* table then all rows in the table with a date less than the 11th may be aggregated by Attribute Name and Originating Date to the *mis_dw_aggregated_count* table after which the extracted entries in *mis_consolidated_aggregates* may be deleted.

For the purposes of measurement and aggregation, the working day will be assumed to be from 20:00 to 20:00. Any file received after 20:00 will be assumed to have been received on the following day.

6.5.5 Batch Timeliness PAS to CAPS

The MIS extract is required to measure whether information regarding Payment Expiry, Encashments and Changes of NPO is delivered to the Benefits Agency within the timescales laid-down in the SLA's.

- We must notify CAPS of payments that have expired by 03:00 on the second day after the last day of payment validity. See System Parameter SLA-PAYMENT
- We must notify CAPS of all encashments by 18:00 on the day after the day of encashment. See System Parameter SLA-ENCASH
- We must notify CAPS of Changes of NPO by 03:00 on the day following the NPO change. See System Parameter SLA-NPO

The measurement of timeliness of information from PAS to CAPS will also utilise the ICMF and OCMF control files which are described in the previous section. The following table shows the subset of file types which are passed across the PAS-CAPS interface which are of interest for the purposes of SLA measurement.

File Type	Description
004	Expired Payment Authorisation
005	Encashments
010	Changes of NPO

The following aggregated totals will be provided to the Data Warehouse:

File Type	Total Transactions	Total Not Notified on Time
004	Tndec	Fndec
005	Tnden	Fnden
010	Tndco	Fndco

FUJ00117516

The *data* files which are passed from PAS/CMS to CAPS will contain a mixed set of transactions. Some of these transactions may have met the SLA's and some which may not. The ICMF/OCMF control files only give us the delivery date/time of the *data* file as a whole and cannot provide the necessary transactional information. In order to extract this information, PAS will write to *file_contents* the counts of records by date (Expiry/Encashment/Change of NPO) as the *data* files are being created.

The MIS extract will process the OCMF files and identify those records which are of the above file types. The extract is only interested in the following:

Field Position	Name	Picture	Description
3-7	Service Id	X(5)	The first 4 characters are the Service Identifier and the last character is the Returning Service Number
8-10	File Type	999	The File Type
11-14	Sequential No	9999	A Sequential No used by CAPS & PAS/CMS to ensure delivery of all *data* files, error reports and control matching files by checking that all numbers within the sequence are accounted for
62-75	Initial Status Change Time	9(14)	The time when the recipient accepted the existence of the data file

When a row in the OCMF file is identified, the Service Id, File Type and Sequential No will be used to select all rows in *file_contents* with the same key. By comparing the actual delivery date from the OCMF file with the *relevant_date* (the date of expiry/encashment/notification) in the *file_contents* table, the extract can determine whether the SLA has been met and aggregate the count of records from the *file_contents* table directly to the Aggregate Staging table.

Each row in *file_contents* will be deleted once the aggregates have been extracted.

Aggregated counts are written to *mis_dw_aggregated_count*.

6.5.6 PUN Production

A CMS process (CMSC201) will write PUN counts directly to *mis_dw_aggregated_count* table with the attribute 'Pic'.

6.5.7 Cardholder End of Interest

This extract reports 12 summary counts of cardholders who have received an 'End of Interest' notification within the period (assumed to be one month).

The counts will be produced monthly for each of the three specified agencies. These agencies have been mapped (hard-coded) to specific card types, and any change to card type reference data is assumed to be via a CP (Change Procedure).

FUJ00117516

Card Type	Agency
1	BA (Benefit Agency)
2	WPA (War Pensions)
3	NISSA (Northern Ireland Social Security))

Details of the counts and their attribute names are shown in the table below:

	BA	WPA	NISSA
End of Interest CAPS Initiated			
Card Stopped	EOICSB	EOICSW	EOICSS
Card Not Stopped	EOICNSB	EOICNSW	EOICNSS
End of Interest Pathway Initiated			
Card Stopped	EOIPSB	EOIPSW	EOIPSS
Card Not Stopped	EOIPNSB	EOIPNSW	EOIPSS

However, because:

1. All cards are stopped upon receipt of End of Interest notification

and 2. Pathway initiated End of Interest is not implemented in release 2

all counts will be zero except for EOICSB, EOICSW and EOICSS.

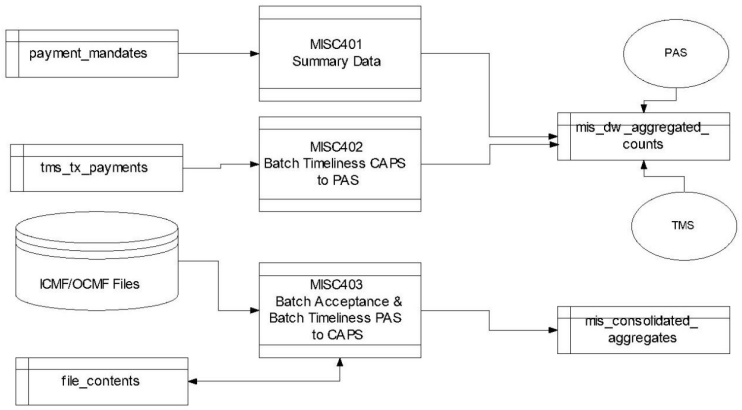
The extract will access cardholder_events rows with a status_code of 'NINT' (within the current month), and classify by card type.

Cardholders that are no longer of interest are notified via a CAPS data feed. Therefore this extract should be run after the feed has been loaded into the PAS/CMS database (Process Cardholders End of Interest - CMSC109). The extracted data is loaded into table mis_dw_aggregated_count with attribute values as shown in the table.

6.5.8 Aggregated Data Extract Data Flow Diagram

The following diagram shows the flow of information between the aggregated data extract processes and the underlying data.

FUJ00117516



7

FUJ00117516

Data Definitions

7.1 Source Data

Most of the source data is available from within the PAS/CMS database although some information is provided in textual format from CAS. The data definitions for source data is defined elsewhere and not within the scope of this document. It is, however, useful to describe which tables of information are accessed and how/when these tables are populated and purged.

7.1.1 BA Profiles

No dependency, this is reference data that is delivered to the Data Warehouse in its most recent form.

7.1.2 Benefit Offices

No dependency, this is reference data that is delivered to the Data Warehouse in its most recent form.

7.1.3 Benefit Types

No dependency, this is reference data that is delivered to the Data Warehouse in its most recent form.

7.1.4 Calls

The calls table is populated by the helpdesk which runs 24 hours per day.

The helpdesk will use a synonym to determine the physical table to which it is writing. Prior to extracting calls to the DW, the extract should switch the synonym to point to an alternative table so that the original table can be extracted in full and eventually truncated.

7.1.5 Cardholder Events

'Not of interest' events are loaded by CMSC109 – Process cardholders End of Interest.

7.1.6 Cards

Cards are created by 'Produce Card & PUN Orders' CMSC201. These are created well in advance of the card being active (typically 2-4 working days) and therefore well in advance of the extract requiring the information.

Card details are purged occasionally once there are sufficient redundant rows to make a purge worthwhile (redundant rows are identified by the non-existence of cardholder information). The process which performs this evaluation and actions the purge is SUPC309. The Extract process is only processing this table for active cards and the purge is therefore not an issue.

FUJ00117516

7.1.7 Cardholders

Cardholder information is not required until they have an active card. Since an active card must have cardholder information, it can be guaranteed that the cardholder row is available.

Cardholder information is purged occasionally once it has accumulated sufficient redundant rows to make a purge worthwhile (Cardholder not of Interest for 90 days or more). The process which performs this evaluation and actions the purge is SUPC307. The Extract is only processing this table for active cards and the purge is therefore not an issue.

7.1.8 Encashed Pay Exceptions

Populated by the process 'Validate Encashed Payments' PMSC302 run daily

Purged by process SUPC314 daily. The exceptions are retained for a time defined by a system parameter which gives operators a chance to view and act on the exceptions.

7.1.9 Encashments & Encashed Payments

This is populated throughout the POCL Core Day by TMS agents.

Encashments are required to be kept for a period of approx. 3 days. Encashment tables are rotated on a daily basis and the current days table is pointed to by the synonym tms_rx_encashments. This is moved to point to the next-days table (suitably truncated) by process 'Encashments Housekeeping' SUPC305 after all overnight encashment processing has completed.

7.1.10 File Contents

Created by processes which pass data from TMS to CAPS; Payment Expiry Process PMSC301, Encashments PMSC310 and NPO Change CMSC110.

Purged by a MIS Extract Process

7.1.11 Mis_dw_aggregated_count

Populated by both PAS/CMS and Extract Processes

PAS/CMS:

MIS Rep 10 Contingency Payments - PMSC310
D08 Rep 2 Payment Authorisation Next Day - PMSC215
D08 Rep3 Payment Stops Non-Urgent - PMSC403
D08 Rep4 Personal Details Notification - PMSC108
D08 Rep 5 End of Interest Notification - PMSC108
D08 Rep 6 Payment Authorisation Regular - PMSC215

The eventual unload to the data warehouse will perform the purge

7.1.12 Mis_dw_card_events

Populated By

Cards Impounded - CMSC115 Service Card Events

FUJ00117516

Customers With Many Lost Cards - CMSC115
Customers With Many Stolen Cards - CMSC115
Customers With Reminder PUN - CMSC115

Purged by the MIS Extract

7.1.13 Mis_dw_unrev_npo

Populated by a new PAS process TBA

Purged by the Extract process which unloads the information to the data warehouse

7.1.14 Mis_inactivated_cards

Populated by deactivated Cards - CMSC115 Service Card Events

Purged by an extract process will run daily and aggregate the information to mis_dw_cdd.

7.1.15 Mis_new_cardholders

Populated by New Customer Details (batch and online). Updated by Card Event Partitioning.

7.1.16 Mis_ordered_cards

Populated by ordered Cards - CMSC201

Purged by the extract process which populates *mis_dw_cdi*

7.1.17 Payment Payees

Payment payees are structured in exactly the same manner as Payments (see below) and are processed in the same manner by the same process.

7.1.18 Payment Events

Payment events may be purged once the associated payment is purged by SUPC301. The population of this table is not of interest to the MIS Extract.

7.1.19 Payment Mandates

Payment mandates are contained within one of two payment mandate tables (A or B) which flip-flop on a daily basis. A synonym *payment_mandates* points to the current table. The process which adds new mandates or updates mandates with the most recent payment details is PMSC205 which runs near the end of the overnight batch schedule.

7.1.20 Payments

The payments table has to be enriched on a daily basis with both the new authorised payments which have been received and with encashment information coming back from the counters. To perform this, the two sets of information are merged with the existing payments table to produce a new payments table. To accomplish this, two payments tables (A & B) exist which flip-flop on a daily basis - the payments table referred to here is a synonym which refers to the current table. Payments which have

FUJ00117516

been enriched with encashment information for more than 48 hours will not be carried forward. The process which performs this is SUPC301.

Any process therefore that wishes to find a payment row for an associated encashment should be guaranteed that the row exists but must run after SUPC301 which enriches the payment information with today's encashment information.

7.1.21 Temporary Token Books

Populated by the Temporary Token Book order process (CMSC401)

7.1.22 Temporary Tokens

Temporary tokens are populated well in advance of when required, but are updated to indicate when they have been used by PMSC314.

Temporary tokens are purged by book (SUPC310) after all TT's have been assigned or stopped and *n* days (as indicated by system parameter temp_token_achive) has passed, and where all TT's have their archive flag set. The archive flag is set when the token is expired or stopped.

This means that tokens could be purged on the same day that the last one expires (CMSC403) or is stopped (HLPF503, CMSC403, CMSC304). *It is therefore essential that temp_token_archive is set to at least 3 days greater than the token expiry period*

7.1.23 Temporary Token Events

Population is performed by the Helpdesk and a range of PAS/CMS processes. Since token events will be processed one day in arrears, the timeliness of population is not an issue to the MIS Extract

Token events are purged along with their associated tokens

7.1.24 Tms_tx_payment_stops

Populated by PMSC403 during the On-line day. This process should complete before the overnight batch schedule starts.

Purged by process SUPC313 at the end of the overnight Batch Schedule

7.1.25 Tms_tx_payments

This table is both created and truncated/dropped by process PMSC215 as part of the overnight batch schedule.

7.1.26 ICMF/OCMF & Online Audit Files

These are delivered from CAS into a pre-agreed directory on the Sequent at an undefined time but understood to be sometime between midnight and 04:00.

FUJ00117516

7.2 MIS Extract Meta-Data

7.2.1 mis_sys_param

Contains general system parameters and working data. See section 0 - 8.3 System Parameters for a full list of table contents.

Name	Type	Description	Comments/Source
Param Type	X(3)	Parameter classification	
Param Subtype	X(8)	Parameter Identifier	
Param Name	X(30)	Description of Parameter	
Param Val	X(100)	Value of parameter	

7.2.2 mis_dw_control

This provides control over the delivery of information to the data warehouse and the purging of the staging tables.

Name	Type	Description	Comments/Source
Process Date	Date	The extract originating date/MIS Extract Date	
Available	Date	The date/time when the control file and all data files are available in the data warehouse repository directory	
Delivered	Date	The date/time when the data was made available to the data warehouse	The time when the lock file was created in the warehouse directory
Purged	Date	Description of Parameter	The date and time when the staging tables were purged

7.2.3 mis_data_files

Containing a complete list of the data files which are to be delivered to the data warehouse. The Data Name is the same 3 character code which uniquely identifies the file to be delivered within the delivery control file. The same 3 character code will also be used in the naming of the staging tables (see below). A full list of the codes can be found in Physical Data Format section of reference [5].

Name	Type	Description	Comments/Source
Data Name	X(3)	The type of data file delivered to the Data Warehouse	This provides part of the interface filename as well as the 'source' element of the interface control file
Staging Table	9	TRUE or FALSE indicating whether a staging table is held for this type of data	Eventually used to ensure that the staging table has been created before unload
Frequency	X	[D]aily, [W]eekly or [M]onthly	Frequency of Data Extract

FUJ00117516

SQL Script	X(999)	The SQL Script necessary to build the Staging Table	Including the Tablespace identifier and any storage clauses but excluding the 'Create Table *Tablename*' clause

7.2.4 mis_daily_files

A row will be held for each *mis_data_files* for each day of delivery. This will provide an audit to ensure that all tabular data has been passed to the Data Warehouse.

Name	Type	Description	Comments/Source
Data Name	X(3)	The type of data file delivered to the Data Warehouse	This provides part of the interface filename as well as the 'source' element of the interface control file
Date	Date	The date of Data Extract	
Staging Table	9	TRUE or FALSE indicating whether the staging table holds completed information	Updated by each process which inserts into staging tables
Delivered	9	TRUE or FALSE indicating whether the data file is completed and delivered to the Data Warehouse	Updated by each process which creates data warehouse interface files
Qty Rows	9(7)	The number of records delivered to the data warehouse	

7.2.5 mis_aggregated_data

Contains a complete list of the aggregated attributes which are to be provided to the Data Warehouse on a daily basis.

Name	Type	Description	Comments/Source
Attribute Name	X(8)	An attribute name as defined in the Data Warehouse EPID	
Frequency	X	[D]aily, [W]eekly or [M]onthly	Frequency of Aggregate Production
Type	X	[A]ggregate, [S]ummary	Indicating the type of aggregated count

7.2.6 mis_daily_aggregates

A row will be held for each *mis_aggregated_data* for each day of delivery. This will provide an audit to ensure that all aggregated data has been passed to the Data Warehouse.

Name	Type	Description	Comments/Source
Attribute Name	X(8)	An attribute name as defined in the Data Warehouse EPID	

FUJ00117516

Date	Date	The date of the extract	
Delivered	9	TRUE or FALSE indicating whether the aggregate has been delivered to the Data Warehouse	0 = Not delivered 1 = Delivered 2 = Missing Aggregate
Delivered Timestamp	Date	The date when the aggregate was delivered to the Data Warehouse	The date when the aggregated was written to the staging table

7.3 Intermediate Data

7.3.1 mis_dw_aggregated_count

This table accumulates all aggregated counts.

Name	Type	Description	Comments/Source
Attribute Name	X(8)	An attribute name as defined in the Data Warehouse EPID	See Section 0 6 Application Processes for an individual description per Attribute Name
Attribute Count	9(8)	The number of occurrences of events indicated by Attribute Name	
Process Date	Date	The date on which the count was produced or the date to which the count relates	

7.3.2 mis_consolidated_aggregates

This table is identical in structure to *mis_dw_aggregated_count* and contains any aggregates for which more than one count may be produced for the same attribute in any one day. This acts as a holding area for those aggregates until all counts have been received for a particular day.

7.3.3 mis_npo_changes

Recording, on a daily basis, customers who have changed NPO *x* or more times in the previous *y* months where:

- *x* = System Parameter NPOTIMES
- *y* = System Parameter MPOMNTHS

Name	Type	Description	Source
Customer NINO	X(8)	The NI Number of the customer	*nominated_post_offices.nino*
No of Times	9(3)	The number of times that the customer changed NPO within the last *y* months	derived

FUJ00117516

7.3.4 mis_tt_stock_out_today

This table holds details of DSS offices that have gone out of stock of temporary tokens today.

Name	Type	Description	Source
Benefit Office Code	X(6)	The code of the Benefit Office which is out of stock.	Derived
Date	Date	The date on which the office is out of stock (this will be the date for which the program is running)	Derived

7.3.5 mis_tt_stock_out_prior

This table holds details of DSS offices that have been out of stock of temporary tokens before today, and have not yet had their stock replenished. Data for a DSS office is removed from this table when it's stock of Temporary Tokens is replenished.

Name	Type	Description	Source
Benefit Office Code	X(6)	The code of the Benefit Office which is out of stock.	Derived
Date	Date	The date on which the office is out of stock (this will be a date prior to the date for which the program is running)	Derived

7.4 Staging Tables

Staging tables will provide a temporary area to place grouped data prior to extraction of that data into the Data Warehouse interface files. These tables only exist as stores for information which cannot be easily reproduced in the event of loss of the Warehouse interface data.

7.4.1 mis_dw_cda

A count of the number of cards active per Post Office. See 0 7.5.9 Active Cards Per Post Office which has an identical structure.

7.4.2 mis_dw_cdi

A count of the number of cards ordered by Post Office and Card Type for each day. See 0 7.5.8 Cards Issued Per Post Office which has an identical structure.

7.4.3 mis_dw_cdd

A count of the number of cards deactivated by Post Office and Event Type for each day. See 0 7.5.10 Deactivated Cards Per Post Office which has an identical structure.

FUJ00117516

7.4.4 mis_dw_cr1

Containing card event data sourced from mis_dw_card_events, this table has the same structure as that defined in section 0 7.5.14 Card Events.

7.4.5 mis_dw_cr2

This table has an identical structure to the Data Warehouse file described in 0 7.5.17 Benefit Non-Encashment.

- Populated daily with details of non-encashment of benefit within 4 weeks of due date - Only for means-tested benefits.
- Populated daily with details of non-encashment of benefit within 6 weeks of due date - For means-tested benefits and Specific non-means tested benefits.

7.4.6 mis_dw_cr6

Populated daily with details of encashments made after a payment stop has been received. See 0 7.5.20 Encashments Made After Stop Received

7.4.7 mis_dw_act

Containing all aggregated counts which have been produced in any one day. This has the same structure as *mis_dw_aggregated_count*.

7.4.8 mis_dw_cr5

Recording customers with many changes of NPO. Refer to the data definition in 0 7.5.19 Customers With Many Changes of NPO.

7.4.9 mis_dw_cqx

Recording all new cardholders registered in one month. This table has the same structure as that defined in section 7.5.25 New Cardholders.

7.4.10 mis_dw_bax

Recording any DSS offices that have become out of stock of temporary tokens, or have been replenished in any one day. This table has the same structure as that defined in section 7.5.26 Temporary Tokens Out of Stock.

7.5 Warehouse Interface Files

7.5.1 Help Desk Calls

The following information is extracted daily from the PAS/CMS *calls* table.

Name	Type	Description	Comments/Source
Call Identifier	9(16)	Unique call Id	*call_id*
Call Opened	Date	Date and time when the call was opened	*start_time*

FUJ00117516

Call Closed	Date	Date and time when the call was closed	*end_time*
Resolution Time	9(5)		*resolution_time*
Operator Id	X(30)		*operator_id*
Call Line	X	Incoming line (English, Welsh, BA, POCL). One of E, W, B or P	*call_line_indicator*
Caller PO	X(7)	The post office code for calls on the POCL staff line	*caller_po_code*
Caller BA Office	X(6)	The Benefits Agency code when the call comes through on the BA staff line	*caller_ba_code*
Caller NINO	X(8)		*caller_nino*
NPO	X(7)	The post office code when the call comes through the customer lines and the customer is a cardholder	*caller_nom_po*
Call Type	X(3)	Main call category	*call_type*
Call Subtype	X(3)		*call_subtype*
PAS/CMS Indicator	X	Flag indicating whether the wrap-up is for a PAS or CMS call	*pms_cms_ind*
Action Count	9(2)		*action_count*
DSS Office	X(6)	PO Linked Office code	*benefit_office*
Caller Reference	X(6)		*caller_reference*
DSS Region	X(3)	DSS Region Code	*office_region_code*
Batch Id	X(13)		*tx_batch_id*
Book Id	9(16)		*tx_book_id*
Encashment Id	X(18)		*tx_encashment_id*
Issue Number	X(3)		*tx_issue_no*
NINO	X(8)		*tx_nino*
Nominated PO	X(7)		*tx_nom_po*
Primary Account Number	9(16)		*tx_pan*
Payment Id	X(18)		*tx_payment_id*
Temporary Token Id	X(18)		*tx_tt_id*
Wrap Code	9(8)	The code used to indicate the type of call made	*tx_wrap_action_id*
Session Id	X(16)		session_id

FUJ00117516

7.5.2 On-Line Main Business Transaction

On-Line transactions consist of a main business transaction followed by zero, one or more secondary transactions. The main business transaction will have up to 5 components. The information is obtained from Audit files which are passed from CAS on a daily basis.

Name	Type	Description	Comments/Source
Start Time	9(18)	The time in microseconds since 31st Dec 1899 of when CAS takes control	On-line Audit File. Ref: [3]. Field Position 3-20
Finish Time	9(18)	The time in microseconds since 31st Dec 1899 of when CAS relinquishes control	On-line Audit File. Ref: [3]. Field Position 21-38
Sequential No	9(18)	Sequential transaction identifier	On-line Audit File. Ref: [3]. Field Position 39-56
No of Business Function	9(4)	A description of what constitutes a Business function can be found in [3].	On-line Audit File. Ref: [3]. Field Position 134-137
Error Code	X(5)	The coding of any fatal error occur which may have occurred within the transaction	On-line Audit File. Ref: [3]. Field Position 164-168
Main Component	9(3)	The first component transaction	On-line Audit File. Ref: [3]. Field Position 169-171
Tx Count	9(4)	The number of main business components	On-line Audit File. Ref: [3]. Field Position 172-175

7.5.3 On-Line Transaction Detail

For each main business transaction described above, there will be up to 5 detail transactions which will be provided to the Data Warehouse in the following form.

Name	Type	Description	Source
Sequential No	9(18)	Foreign key to the Main Business Transaction	On-line Audit File. Ref: [3]. Field Position 39-56
Component	9(3)	Scorecard Component	On-line Audit File. Ref: [3]. The first 3 characters of five repeating groups of seven characters within position 169-203.
Count	9(4)	Scorecard Component Count	On-line Audit File. Ref: [3]. The last 4 characters of five repeating groups of seven characters within position 169-203.
Type	X	Flag to indicate whether this is the Main Business Component or a Secondary Component. Value may be M or S.	Derived. The first Component within a transaction will be the Main Business Component

FUJ00117516

7.5.4 Payee Roles

Reference data taken directly from the *payee_roles* table.

Name	Type	Description	Source
Payee Role	9(2)	Unique Role Code	*payee_role*
Role Description	X(45)	Description of the Payee Role	*payee_role_desc*
Classification Code	X(1)	Classification Code	*classification_code*

7.5.5 Benefit Agencies

Reference data taken directly from the *benefit_agencies* table.

Name	Type	Description	Source
Agency Code	9(2)	Unique Agency Code	*agency_code*
Agency Description	X(16)	Description of the Agency	*agency_description*

7.5.6 Benefit Types

Reference data taken directly from the *benefit_types* table.

Name	Type	Description	Source
Benefit Type	9(3)	Unique code for this Benefit Type	*benefit_type*
Benefit Description	X(12)	Description of the Benefit	benefit_description
Agency Code	9(2)	The code of the Agency responsible for this benefit	*NOT AVAILABLE*
Means Tested	X	Flag indicating whether this benefit is means tested	means_tested_ind Y or NULL
Long Description	X(40)	A longer description of this benefit type	*benefit_desc_long*

7.5.7 Call Wrap Codes

Reference data taken directly from the *wrap_enquiry_actions* table

Name	Type	Description	Source
Wrap Code	9(8)	Unique help-desk call wrap-up code	*wrap_action_id*
PAS/CMS Flag	X	Flag indicating whether the wrap-up is for a PAS or CMS call	*pas_cms_ind*
Wrap Description	X(40)	Wrap-up description	*enquiry_action*

FUJ00117516

7.5.8 Cards Issued Per Post Office

A summary of the number of cards issued for each Post Office Daily This data is extracted from one of the staging tables (*mis_dw_cdi)* which in turn is the sum of all *mis_ordered_cards* within one day.

Name	Type	Description	Source
PO Code	X(6)	Post Office FAD Code	*po_code*
Card Design Type	9(2)	Type of this card. Eg. Benefit card, Northern Ireland or War Pensions cards	*card_type*
Issue Reason Code	9(2)	The event type which caused the card to be ordered from De-la-rue	*event_type*
Card Count	9(8)	Count of the number of cards issued during the month for this PO, Design & Reason	The sum of *mis_ordered_cards* for one day
Bilingual Ind	X	Indicating those cards issued in both English and Welsh	*billingual_crd_ind*
DSS Issuing Office	X(6)	The Post Office 'Linked Office'	*linked_office*
Extract Date	Date	The source date of the extract	Derived

7.5.9 Active Cards Per Post Office

A summary of the number of currently active cards issued for each Post Office is produced at the end of each week. This data is extracted from one of the staging tables which in turn is the sum of all cards with a status 'ACT'.

Name	Type	Description	Source
PO Code	X(6)	The Post Office FAD Code	*cardholders.po_code*
Card Count	9(8)	The number of cards active at this point in time associated with this Post Office	*cards* - count of
DSS Issuing Office	X(6)	Assumed to be the linked office	*post_offices.linked_office*
Extract Date	Date	The source date of the extract	Derived

7.5.10 Deactivated Cards Per Post Office

A summary of the number of deactivated cards for each Post Office is produced at the end of each week. This data is extracted from one of the staging tables which in turn is the sum of rows placed in *mis_inactivated_cards*.

Name	Type	Description	Source
PO Code	X(6)	Post Office FAD Code	*po_code*

FUJ00117516

Deactivation Reason Code	9(2)	Event Type which caused deactivation	*event_type*
Card Count	9(8)	The number of cards per Post Office and Deactivation Reason	Sum of *mis_inactivated_cards* with this combination of PO and Reason
Extract Date	Date	The source date of the extract	Derived

7.5.11 Despatch Batches

One row will be delivered to the Data Warehouse for each batch of cards which has attained a status of 'BKD'. This will be instigated by identifying a despatch batch event of status '10' - Batch Reconciled. Delivered daily from table *despatch_batches*.

Name	Type	Description	Source
Batch Id	X(13)	The unique identifier for each batch	*batch_id*
PO FAD Code	X(6)	The PO FAD Code truncated to 6 characters	*po_code*
Card Type	99	The card type as defined in reference data *card_types*	*card_type*
Due Date	Date	The date from which the cards within the batch are valid	*due_date*

7.5.12 Despatch Batch Cards

This contains an aggregated count of cards with each batch and is delivered to the Data Warehouse at the same time as Despatch Batches above. This data is extracted directly from *despatch_batch_cards*.

Name	Type	Description	Source
Batch Id	X(13)	The unique identifier for each batch	*batch_id*
Original Order Date	Date	The date on which an individual card was originally ordered	*original_order_date*
Original Card Event Type	99	The event which instigated the original order of each individual card	*original_card_event_type*
Urgent Indicator	X	Indicating whether the original card order was deemed urgent	*urgent_ind*
Count	9(5)	The count of cards in the batch grouped by all of the above	Derived

7.5.13 Despatch Batch Events

The following events are extracted from *despatch_batch_events* on a daily basis for all batches which have a status of 'BKD':

03 - Batch received at PO

FUJ00117516

07 - Batch Ordered
10 - Batch Reconciled
14 - Delivered to private box number
15 - Delivered to destination address
16 - Collected by recipient from callers office
25 - Despatched from De-La-Rue

Name	Type	Description	Source
Batch Id	X(13)	The unique identifier for each batch	*batch_id*
Event Type	99	The event code as listed above	*event_type*
Event Sequence	9(16)	The sequence of this event for this batch	*event_seq*
Event Timestamp	Date	The date/time that this event was recorded	*event_tsmp*
Status Code	X(3)	The resulting batch status	*status_code*

7.5.14 Card Events

This is an extract of *mis_dw_card_events* which is populated by various CMS and Helpdesk processes to record the following event types:

- Impound Events
- Lost Card Events
- Stolen Card Events
- Reminder PUN Events
- Card Not Collected Events

During the extract, the Linked Office code is added to the data provided in *mis_dw_card_events* from the Post Office reference data.

Name	Type	Description	Source
Primary Account No	9(16)	The primary account number of a card	
Card Issue No	9(3)	The issue number of a card. Appending this to the primary account number uniquely identifies a card	
Event Timestamp	Date	The date/time when this card event took place	
Event Type	9(2)	The type of event	
Event Source	X	The source of the event	PMS, CMS, HELPDESK, SUPP (Support System), PO, POLI (On-Line Interface), PMSRCL (PMS Payment Stop with all Authorisations confirmed recalled)

FUJ00117516

Source Desc	9(2)	A code which indicates the description of the source	1 Customer 2 Customer (Welsh) 3 Benefit Agency 4 PO Counter 5 Relative 6 Friend 7 Associate 8 Public 9 Police 10 Financial Institution 11 Other
Source1_id	X(30)	Entered at the Help Desk as 'Person' - used for audit to identify the individual calling	
Source2_id	X(30)	Entered at the Help Desk as 'Organisation' - used for audit to identify the individual calling	
call_id	9(16)	Unique identifier of the call record for the call made to the Help Desk.	
PO Fad Code	X(6)	The post office identifier	Derived when extracting from *mis_wd_card_events* from *cardholders*
Linked Office	X(6)	Linked office (DSS Issuing Office)	Derived when extracting from *mis_dw_card_events* from Post Office reference data
NINO	X(8)	The NINO of the cardholder	Derived when extracting from *mis_dw_card_events* from *cards*

7.5.15 PO Temporary Closure

The following data is required on a daily basis and is extracted directly from *po_events*.

Name	Type	Description	Source
PO FAD Code	X(6)	The Post Office FAD Code	*po_code*
Event Date	Date	The date when the post office opened or closed	*event_tsmp*
Event Code	X	The event type (Opening or Closure)	*status* This will be one of [O]pen, [T]emporarily closed, [E]mergency closure

7.5.16 PUNS Not Received

This is a direct extract of *mis_repudiated_puns* which is populated by the PAS/CMS Helpdesk for all PUNS reported not received yet the associated card has been collected

Name	Type	Description	Source
Extract Date	Date	The source date of the extract	Derived

FUJ00117516

Cardholder NINO	X(8)	The NI Number of the cardholder	*cardholder_nino*
Card PAN	9(16)	The Primary Account Number of the associated card	*primary_ac_no*
Card Issue No	9(3)	The card Issue Number	*card_issue_no*
Report Date	Date	Date when PUN Reported as Not Received	*event_tsmp*
Card Collection Date	Date	The date when the card was collected	*card_collected*

7.5.17 Benefit Non-Encashment

Extracted daily from *payments* and *benefit_*types.

Name	Type	Description	Source
Payee NINO	X(8)	The NINO of the customer	*payments.customer_nino*
Benefit Type	9(3)	The type of benefit for which the payment was authorised	*payments.benefit_type*
Means Tested Indicator	X	Indicating whether the payment is means tested	*benefit_types.means_tested_ind*
Specific Non-Means Tested Indicator	X	Indicating whether the payment is not means tested but is of other specific interest	*benefit_types. Spec_non_means_tested_ind*
Non-encashment Weeks	9	Number of weeks that this payment has been non-encashed for either 4 or 6 weeks beyond the earliest encashment date	Derived
Payment Id	9(18)		*payments.payment_id*
Due Date	Date	Earliest encashment date	*payments.earliest_encash_date*
Amount	9(8,2)		*payments.encashable_amt*

7.5.18 Changes of NPO

Produced daily, this records Individual Instances of Change of Nominated P.O. not Reversed Within 6 Weeks., where there is encashment of means-tested or specified non-means tested benefit whose date of availability is after change of NPO and there is no notification of change of address received within 6 weeks.

The data is extracted directly from *mis_dw_unrev_npo* which is populated from PAS/CMS processes.

Name	Type	Description	Source
Customer NINO	X(8)	The NI Number of the customer	
Timestamp	Date	Date when the conditions for	

FUJ00117516

		reporting were met	
Old PO	X(6)	Original Post Office FAD Code	
New PO	X(6)	New Post Office FAD Code	
Source	X	The source of the change of address details	C = CAPS O = PO (via TPS) P = POLI NULL = Other

7.5.19 Customers With Many Changes of NPO

Recording, on a monthly basis, customers who have changed NPO *x* or more times in the previous *y* months where:

- *x* = System Parameter NPOTIMES
- *y* = System Parameter MPOMNTHS

Name	Type	Description	Source
Customer NINO	X(8)	The NI Number of the customer	
No of Times	9(3)	The maximum number of times that the customer changed NPO	Derived
Extract Date	Date	The source date of the extract	Derived

7.5.20 Encashments Made After Stop Received

Produced daily from *encashed_pay_excptns* and *tms_rx_encashments*

Name	Type	Description	Source
Customer NINO	X(8)	The NI Number of the customer	*tms_rx_encashments. customer_nino*
Encashment Id	X(18)	The unique identifier of an encashment	*encashed_pay_excptns. excptn_detail(27,44)*
Stop Timestamp	Date	The date/time when the stop was received	*payment_events.event_tsmp*
PO Code	X(6)	The PO FAD Code	*tms_rx_encashments.po_code*
PO Clerk Id	X(6)	The Id of the Post Office Clerk	*tms_rx_encashments.po_clerk_id*
Amount	9(8,2)	The amount of the Authorised Payment which was stopped	*encashed_pay_excptns. excptn_detail(19,26)*
Milk Tokens Issued	X	Indicating whether there were milk tokens issued with the Authorised Payment	*tms_rx_encashments.token_type_1*
Casual Agent Flag	X	Indicating whether the encashment was made by a Casual Agent	*tms_rx_encashments.casual_agent*
Foreign Encashment	X	Indicating whether the encashment was made at a foreign Post Office	*tms_rx_encashments foreign_po_ind*

FUJ00117516

Temporary Token	X	Indicating whether the encashment was made using a Temporary Token	*tms_rx_encashments. temp_token_ind*
Payment Id	9(18)	The payment Id of the stopped payment	*encashed_pay_excptns. excptn_detail(1,18)*
Encashment Timestamp	Date	The date/time when the payment was encashed	*tms_rx_encashments. encashment_tsmp*
Extract Date	Date	The source date of the extract	Derived

7.5.21 Temporary Tokens Issued

Delivered to the Data Warehouse on a daily basis giving counts of all temporary tokens Issued, Impounded and Unused at Expiry Date. The data is sourced from *temporary_tokens*, *temporary_token_books*, *temporary_token_events* and *benefits_offices*.

Name	Type	Description	Source
DSS Issuing Office	X(6)	The benefits Agency Office which issued this Temporary Token	*temporary_token_books. holding_ba_office*
DSS Region	X(3)	The Office Region Code	*Benefits_offices. Office_region_code*
Tokens Assigned	9(6)		Derived
Tokens Impounded	9(6)		Derived
Tokens Unused	9(6)		Derived
Extract Date	Date	The source date of the extract	Derived

7.5.22 Random Selection of Encashments

The following data is required daily for encashment records "randomly" selected. All information comes directly as a single record from PAS/CMS table *random_encashments*

Name	Type	Description	Source
Record Type	99	Identifies record type in file	
Beneficiary NINO group	X(9)	The NINO of the customer who has entitlement to this payment	
Record Creation Date	9(8)	The record date.	
Record Creation Time	9(6)	The record time.	
Encashment Identifier	X(18)	The unique identifier for the encashment event.	

FUJ00117516

Post Office Identifier	X(6)	The post office at which the client made this encashment. Corresponds to "encashment location" in CAPS. (1st 6 digits of FAD code).	
Post Office Clerk ID	X(6)	The ID of the clerk encashing the payment.	
Encashment Amount	9(8)	The total amount of cash given to the client in this encashment.	
Total Issued Token Amount	9(5)	The total number of tokens issued as part of this encashment.	
Encashment Date	Date	The date on which this encashment took place.	
Encashment Time	Date	The time at which this encashment was completed.	
Payee NINO group	X(12)	The NINO of the customer who collected this payment.	
Casual Agent Flag	X	Y if casual agent made this encashment, N otherwise.	
Keyed Card Details Flag	X	Y if encashment enabled by keying card number, N otherwise.	
Foreign encashment flag	X	Y if encashment counts towards customer's foreign encashment limit, N otherwise.	
Encashed payment group count	99	The number of encashed payment records relating to this encashment.	
Encashed tokens group count	99	The number of encashed tokens records relating to this encashment.	
Temporary token encashment group count	99	The number of temporary token encashment groups relating to this encashment. (Where present, indicates that a temporary token was used to make the transaction rather than a payment card).	
Group Type	99	Identifies this payment group type within this record.	
Group Sequence Number	99	The sequence number of this payment group within the encashment record.	
Payment ID	X(18)	The unique identifier for the payment that has been encashed.	
Encashed amount	9(8)	The encashed amount of this payment.	

FUJ00117516

Benefit Type[2]	9(3)	The type of benefit for which this encashment was made.	
Running Count	9(11)	Running count of encashments processed	
Running total	9(18)	Running total value of encashments processed	
Penny hit	9(14)	Penny hit within selected encashment (i.e. derived from random number that determined selection).	
Trailer Record Type	99	Identifies this trailer record type within this file.	
File Type	9(3)	Identifies the type of file.	
Total Record Count	9(8)	Count of all detail records with the [CAPS transfer] file.	
Group Type 22 count	9(8)	Count of all groups of type 22 within this [CAPS transfer] file.	
Group Type 23 count	9(8)	Count of all groups of type 23 within this [CAPS transfer] file.	
Group Type 31 count	9(8)	Count of all groups of type 31 within this [CAPS transfer] file.	
Total of encashment amount	9(14)	Total of all encashment amount fields from the encashment records.	
Total of total issued token count	9(6)	Total of all total issued token count fields from encashment records.	
File creation date[3]	Date	Must have *exactly* the same value as *all* *Payments_encashment_records.record_creation_date* transferred today.	
MUS Cell Size	9(14)	The size of the cell used for MUS sampling (as specified by NAO).	
Extract Date	Date	The date of extraction to the data warehouse	

7.5.23 Benefit Offices

Reference data taken directly from the *benefit_offices* table.

Name	Type	Description	Source

[2] To be added by CMS/PAS to CAPS record before transfer to Data Warehouse.

[3] To be added by CMS/PAS to CAPS record before transfer to Data Warehouse.

FUJ00117516

Benefit Office	X(6)		*benefit_office_code*
Post Code	X(8)		*postcode*
Office Name	X(27)		*office_name*
Office Region Code	X(3)		*office_region_code*
Country Code	X(3)		*country_code*
Expiry Date	Date		*expiry_date*
Redirection Office Number	9(6)		*redirection_office_no*
Office Open Date	Date		*office_open_date*
Office Closure Date	Date		*office_closure_date*
Parent Office	9(6)		*parent_office*
Office Type	9(2)		*office_type*
Office Location Type	9(1)		*office_location_type*
Profile Id	9(3)		*profile_id*

7.5.24 BA Profiles

Reference data taken directly from the *ba_profiles* table.

Name	Type	Description	Source
Profile Id	9(3)		*profile_id*
Reorder Quantity	9(8)		*reorder_quantity*
Reorder Level	9(8)		*reorder_level*
Initial Order	9(8)		*initial_order*
Profile Description	X(40)		*profile_desc*

7.5.25 New Cardholders

Extracted from mis_new_cardholders and cardholders

Name	Type	Description	Source
Primary Account Number	9(16)		*primary_account_no*
Card Issue Number	9(3)		*card_issue_no*
Card Event	9(2)		*card_event*

FUJ00117516

Card Type	9(2)		*card_type*
Cardholder NINO	X(8)		*cardholder_nino*
DLO Indicator	X		*invalid_address_marker*

7.5.26 Temporary Tokens Out of Stock

Extracted from temporary_token_books

Name	Type	Description	Source
DSS Office Code	X(6)		*holding_ba_office*
Event Date	Date		*Event_date*
Event Type	X	Either: O: Out of Stock or R: Replenished	*Derived*

FUJ00117516

8 Appendices

8.1 Data Volumes

Table Name	Rows	Row Size	Total Size	Comments
mis_aggregated_data	53	17	901	
mis_consolidated_aggregates	159	30	4770	The total number of aggregates * 3
mis_daily_aggregates	212	32	6784	The total number of aggregates * 4
mis_daily_files	116	28	3248	The total number of data files * 4
mis_data_files	29	1013	29377	
mis_dw_act	53	30	1590	Assume a maximum of 4 tables
mis_dw_aggregated_count	53	30	1590	
mis_dw_bax	100	15	1500	Assume 4 Tables
mis_dw_card_events	16000	142	2272000	Assume 10% card events are impound events
mis_dw_cda	20000	35	700000	Assuming one per post office. Assume 4 Tables
mis_dw_cdd	30000	31	930000	Stop events per P.O. per event. There are 29 card stop events and 20000 POs. Not all PO's will create a deactivate event each day but some may record many types. Difficult to estimate the number of rows Assume 4 Tables
mis_dw_cdi	60000	40	2400000	Cards ordered per P.O. per Card Type. There are 3 Card Types and 20000 Post Offices. Assume 4 Tables
mis_dw_control	3650	39	142350	This provides for 10 years of operation
mis_dw_cqx	1820000	32	58240000	Based on Pathway Maximum Rollout figures. Assume 4 Tables
mis_dw_cr1	16000	139	2224000	Assume 4 Tables * mis_dw_card_events
mis_dw_cr2	30000	59	1770000	Benefits non-encashment is difficult to evaluate. Assume 0.5% of payments remain non-encashed (an unlikely high figure). Assume 4 Tables
mis_dw_cr5	2400	24	57600	Customers with many changes of NPO - difficult to evaluate. Assume 0.01%.Assume 4 Tables
mis_dw_cr6	100	108	10800	Encashments after stop received. Unlikely - Assume 4 tables
mis_dw_unrev_npo	100	30	3000	Changes of NPO not reversed. Unlikely
mis_inactivated_cards	60000	25	1500000	As mis_dw_cdd
mis_npo_changes	7200	15	108000	A compilation of changes of NPO within one month.
mis_ordered_cards	30000	32	960000	As mis_dw_cdi
mis_sys_param	20	168	3360	
mis_tt_stock_out_today	500	15	7500	
mis_tt_stock_out_prior	500	15	7500	

FUJ00117516

8.2

FUJ00117516
FUJ00117516

Function/Table Cross Reference

8.3

PAS/CMS MIS Data Extract
Function/Table Cross Reference

Table/Process	MISC101	MISC102	MISC103	MISC104	MISC201	MISC202	MISC203	MISC204	MISC205	MISC206	MISC207	MISC208	MISC209	MISC301	MISC302	MISC303	MISC304	MISC305	MISC306	MISC307	MISC308	MISC309	MISC401	MISC402	MISC403	MISC404
ba_profiles		The number of calls tables held				R				Initially set to 3																
beneficiaries																							R			
benefit_agencies						R																				
benefit_types		The currently active *calls* table								1, 2, 3 etc								R								
benefits_offices						R				R												R				
calls_n					RD																					
cardholders		The last date when the synonym to the *calls* table was switched								A date in the form YYYYMMDD			R		R		R				R					
cards													R								R					
despatch_batch_cards												R														
despatch_batch_events												R														
despatch_batches												RU														
encashed_pay_excptns		Start of POCL Core Day.								Known as 'Start of POCL Core Day' this is the SLA measurement time for delivery of transactions to the Correspondence servers. Measured in number of hours beyond midnight, this will initially be set to 8.									R								
encashed_payments_n																			R							
encashments_n																			R							
mis_aggregated_data	R	R																					R	R		
mis_consolidated_aggregates		RD								Measured in number of hours beyond midnight, this will initially be set to 8.															C	
mis_daily_aggregates	C	RU	D																							
mis_daily_files	CR	RU	RD		U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U					
mis_data_files	R	Start of POCL Core Day+N		R						This is the second SLA measurement time for delivery of transactions to the Correspondence servers. Measured in number of hours beyond midnight, this will initially be set to 11.												C				C
mis_dw_act			CR	D																						
mis_dw_bax			R	D																		C				
mis_dw_aggregated_count			CD																		C		C	C		C
mis_dw_card_events																	CD									
mis_dw_cda			R	D											C											
mis_dw_cdd			R	D												C										
mis_dw_cdi			R	D												C										
mis_dw_cqx			R	D	Acceptance Cut-off Time					The time after which files of transactions from CAPS will not be measured for conformance to SLA's. Initially set to 20																
mis_dw_control			CRU	UD																						
mis_dw_cr1			R	D												C										
mis_dw_cr2			R	D													C									
mis_dw_cr5			R	D																C						
mis_dw_cr6			R	D	File Acceptance Time					The number of hours after FCUTOFF when PAS/CMS should have acknowledged files received from CAPS. Initially set to 2.						RD			C							
mis_dw_unrev_npo									RD																	
mis_new_cardholders																					R					
mis_inactivated_cards																RD										
mis_npo_changes																				CD						
mis_ordered_cards															RD											
mis_repudiated_puns													RD													
mis_sys_param		Transaction Acceptance Time	R	R	U	RU	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		R	
mis_tt_stock_out_prior										The number of hours after FCUTOFF within which PAS/CMS should have accepted files of transactions received from CAPS - Initially set to 6.													CR				
mis_tt_stock_out_today																							CR			
nominated_post_offices																				R						
payee_roles		Payment Expiry Notification				R				The Number of hours after payment expiry within which CAPS should have notified. Initially set to 27.								R								
payment_events																										
payment_mandates																							R			
payments																		R								
po_events								R																		
post_offices		Encashment Notification								The Number of hours after encashment within which CAPS should have notified. Initially set to 18.																
system_codes	R	R	R		R	R	R	R	R													R	R	R	R	
system_parameters																							R	R		
temporary_token_books																						R				
temporary_token_events																										
temporary_tokens		Change of NPO Notification								The Number of hours after change of NPO within which CAPS should have notified. Initially set to 3.																
tms_tx_payment_stops																								R		
tms_tx_payments																								R		
wrap_enquiry_actions						R																				
DWH REPOSIT		The data warehouse								The root repository directory where all extracted files are placed initially set to																

FUJ00117516

		repository directory	'/bvnw01/tps/dw'
DWH	SYSTEM	The PAS/CMS system within the repository	This is appended to the Repository directory to form the root directory where PAS/CMS extracted files are placed. Initially set to 'cms'.
LOC	REPOSIT	The fallback repository directory	The root repository directory where all extracted files are placed in the event of fallback. Initially set to '/bvnw01/tps/fallback_dw'
LOC	SYSTEM	The PAS/CMS system within the fallback repository	This is appended to the fallback Repository directory to form the root directory where PAS/CMS extracted files are placed. Initially set to 'cms'.
MIS	PLATFORM	Indicating whether the MIS Extract is delivering to the warehouse platform or the local platform	'DWH' or 'LOC'
MIS	ONLINE	The directory where On-line transactions are delivered	
MIS	CMF	The directory where inward and outward control matching files are delivered	
MIS	NPOPURGE	The number of days that rows will be held in mis_dw_unrev_npo	Initially set to 5
MIS	PUNPURGE	The number of days that rows will be held in *mis_repudiated_puns*	Initially set to 5
MIS	WEEKEND	The number of the day in the week when weekly tasks should run. This equates to Oracles interpretation of a weekday 1=Sunday, 7 = Saturday	Initially set to 1
MIS	NPOTIMES	The number of times that a customer must change NPO within NPOMNTHS before being extracted	Initially set to 6. This should only be changed immediately after month-end
MIS	NPOMNTHS	The duration of history evaluated when checking for many changes of NPO	Initially set to 12. This should only be changed immediately after month-end
MIS	DELIVERY	The number of minutes after midnight on the date of extract when the extract must be delivered to the data warehouse	Initially set to 120
MIS	MISPURGE	The number of days that staging tables will be retained on the system after delivery to the data	Initially set to 3

FUJ00117516

		warehouse	
MIS	ENCPURGE	The number of days that random encashment rows will be retained on the system after delivery to the data warehouse	Initially set to 3
MIS	INCPURGE	The number of days that rows will be retained within MIS_DW_INFR_NPO_CH after delivery to the data warehouse	Initially set to 3