Classification: Official

NHS England

The Person_ID Handbook

A user guide to *Person_ID* and its creation via the Master Person Service, with applications to HES data sets

Document Name	**The Person_ID Handbook for HES users**		
Project / Programme	**MPS Enhancement**	Project	**MPS Enhancement**
Project Manager	**Giulia Mantovani**	Status	**Final**
Owner	**Rupert Chaplin**	Version	**2.0.0**
Author	**Data Science Team**	Version issue date	**17/01/2024**

Document management

Revision History

Version	Date	Summary of Changes
V0.0.1	29/03/2022	Restructured flow, cleaned comments, accepted edits.
V0.0.3	14/04/2022	Added chapters, edited document.
V0.0.4	28/04/2022	Added chapters, edited document.
V0.1.0	29/04/2022	Revision and edits to ready the doc for first external review
V0.1.2	12/05/2022	Embedded external reviewers' feedback. Ready the document for CR, DG and users review
V0.3.1	15/06/2022	Resolve feedback from users and merge with changes from DG's review.
V0.4.0	16/06/2022	Final version ready for sign-off by Chris, pending Azhar's team revision
V0.4.1	30/6/2022	Revised by Rupert, minor adjustments to text and tables
V1.0.0	04/07/2022	Ready for publication
V1.0.1	05/07/2022	Corrected contact email address
V1.0.2	12/07/2022	Added a navigation page and modified page numbering
V1.0.3	20/07/2022	Removed broken links, updated wrong values in case studies 8 and 10, added eligibility criteria in chapter 3.5. Amended error regarding partial DOB in chapter 3.5 (paragraph concerning the blocking stage of the algorithm).
V1.0.4	16/12/2022	Updated description on handling sensitive flags (pages 5 and 23). Added note on 'perfect matching' in MPS (page 1). Added description of error codes (pages 41-42). Added comments on superseded *NHS numbers* (page 26). Various other typographical corrections
V2.0.0	17/01/2024	Addition of details for alphanumeric and algorithmic trace chapters Addition of reference to MPS diagnostics user guide Addition of an appendix on Soundex Change to the new NHS England template Addition of explanation of which assets can populate the MPS record bucket Other minor adjustments

Reviewers

This document must be reviewed by the following people:

Reviewer name	Title / Responsibility	Date	Version
Rupert Chaplin	Head of Data Science	27/06/2022	V0.4.0
David Hallam	Developer in DPS	03/05/2022	V0.1.0
Azhar Nisar	Principal Systems Engineer in Spine Core	16/06/2022	V0.1.0
Dominic Gair	HES IAO	19/05/2022	V0.2.0
Richard Bradley	Webpage owner	27/06/2022	V0.4.0
Rupert Chaplin	Head of Data Science	19/12/2022	V1.0.4
Claudiu Dumitrascu	Spine Data Engineer	12/12/2023	V2.0.0
Chris Clarke	Spine Lead Data Engineer	12/12/2023	V2.0.0
Rupert Chaplin	Assistant Director Data Science	02/01/2023	V2.0.0

Approved by

This document must be approved by the following people:

Name	Signature	Title	Date	Version
Chris Roebuck		Chief Statistician	01/07/2022	V1.0.0
Stephen Koch		Executive Director, Platforms	01/07/2022	V1.0.0
Chris Roebuck		Chief Statistician	21/12/2022	V1.0.4

Document Control:

The controlled copy of this document is maintained in the NHS England corporate network. Any copies of this document held outside of that area, in whatever format (for example paper, email attachment), are considered to have passed out of control and should be checked for currency and validity.

Contents

Executive summary and outline of the document

The *Person_ID* is a unique patient identifier used by NHS England with the objective of standardizing the approach to patient-level data linkage across different data sets.

This handbook aims to provide users of the *Person_ID* in the Hospital Episode Statistics (HES) databases with supporting documentation on what the *Person_ID* is, how it is derived via the Master Person Service (MPS), how the data flows between services (Data Processing Services (DPS) and Spine), and how to interpret the output information associated with the *Person_ID*.

*Person_ID*s are provided in many data sets available in NHS England including HES, and are derived from the outputs of MPS. For security and privacy reasons many users might have visibility of the tokenised version of the *Person_ID*, which provides an extra level of patient confidentiality.

MPS takes certain demographic information contained in a person's health and care records and matches it to their unique *NHS number* to confirm their identity. The collection of all *NHS number*s and patients' demographic information is contained in the Personal Demographics Service (PDS) data set.

Like any data linkage method, MPS can provide non-perfect matching. There are risks of both failing to match a record (false negative) and matching to a record incorrectly (false positive). The performance of MPS is determined by both the algorithm itself and the quality of incoming data.

MPS operates in the same way for all data sets and is not tuned to any particular use case. For example, where records reliably have accurate NHS numbers attached, MPS will provide a correct match with high confidence. Where solely relying on other personal identifiers (such as name, postcode, gender or date of birth), which may be incomplete, inconsistently recorded or duplicated across the population, the algorithm will be less able to return a correct match in all cases.

Mature health datasets, where identity is typically validated in a healthcare setting at point of recording (such as HES), have higher levels of matching accuracy through MPS for most records. Performance for other datasets may be variable.

Where a perfect match of *NHS number* and *date of birth* cannot be found between a record of interest and any of the PDS records, more complex algorithms are used to compare partial demographic information to identify the most likely PDS record corresponding to the query

record. These algorithms are referred to as alphanumeric and algorithmic trace, but in HES only the latter is used. In the algorithmic trace step, the single queried record is compared to all records in PDS. The comparisons involve some demographic information (*date of birth, name, gender* and *postcode*) and are scored based on similarity. If the similarity is deemed acceptable, the matched record is returned. Otherwise, the algorithm proceeds to look for similarities between the record of interest and some previously unmatched records, stored in the MPS record bucket, a separate data set.

The *Person_ID* is therefore one of *NHS number* from PDS, *MPS_ID* from the MPS record bucket or a *one-time-use ID*, depending on if and where a match was found.

The rest of the document is structured as follows:

- the abbreviations and glossary section provides a useful collection of terms to refer to throughout the document
- chapter 1 explains what the *Person_ID* is and provides details on the scope of this document
- chapter 2 explains how the *Person_ID* is generated and how it is used in the context of the HES data set
- chapter 3 provides a more detailed technical explanation of the algorithms behind the matching logic
- chapter 4 shows specific empirical examples of how a *Person_ID* is matched
- finally, chapter 5 contains additional information helpful to the *Person_ID* users.

Abbreviations and glossary

The following table contains a quick reference for abbreviation terms across the document.

Term	Definition	Meaning
DOB	Date of Birth	Date of Birth using the format YYYYMMDD.
DPS	Data Processing Services	A set of secure technologies and processes that enable NHS England to collect, process and access data. See DPS.
GP	General Practitioner	General practitioners (GPs) treat all common medical conditions and refer patients to hospitals and other medical services for urgent and specialist treatment.
HES	Hospital Episode Statistics	Hospital Episode Statistics (HES) is a database containing details of all admissions, A&E attendances and outpatient appointments at NHS hospitals in England. See HES.
MESH	Message Exchange for Social Care and Health	The Message Exchange for Social Care and Health (MESH) is a secure file transfer service used across health and social care organisations. See MESH.
MPS	Master Person Service	A system that enables data linkage between data sets through *Person_ID* by tracing and verifying key identifiers against the Personal Demographics Service (PDS). See MPS.
MPSaaS	Master Person Service as a Service	A service that allows the request of a Person_ID outside of the normal data set processing pipelines in DPS. This is a service offered by DPS which relies on MPS.
NBO	National Back Office	The NBO provides a national data quality service. It is responsible for the management of NHS numbers and PDS records, investigation and resolution of data quality incidents on PDS demographic records, and the provision of a Tracing Service to approved organisations. See NBO.
PDS	Personal Demographic Service	PDS is the national electronic database of NHS patient details, which holds about 80M records. The demographic details are normally updated when patients visit their GP but may also be updated by other healthcare professionals. See PDS.
UPRI 1	Unmatched person record identifier 1	Another term used for *MPS_ID* – the identifier generated if a match cannot be found in PDS but a record can be linked or created in the MPS record bucket.
UPRI 2	Unmatched person record identifier 2	This is the identifier generated if neither an *NHS number* nor *MPS_ID* (*UPRI* 1) exists.

The following table contains a quick reference for common terms across the document.

Algorithm	An algorithm is a set of well-defined instructions used to solve a class of specific problems. In this context, the algorithm is defined to undertake data linkage, the task of finding records in a data set that refer to the same entity across different data sources.
Algorithmic trace	Algorithmic trace is the final stage in MPS to match records with PDS. For each query record, a set of PDS candidate records are identified by blocking. Each PDS candidate is then scored. The highest scoring candidate record is chosen as the matching record.
Alphanumeric trace	Alphanumeric trace is the second trace step in MPS. The minimum required fields are *family name*, *year of birth* and *gender*, but as HES does not contain names, this step is skipped for HES data sets.
Blocking	Blocking divides datasets into sections, called blocks, to reduce the number of comparisons that need to be conducted to find the linked PDS record.
Cross-check trace	Cross-check trace is the first and simplest tracing step, which can be used when *NHS number* and *DOB* are present. This appears twice, once against the cached version of PDS within DPS and a more complex version is present in Spine against the live version of PDS.
Data linkage	Data linkage is the task of finding records in a data set that refer to the same entity (that is, patient in the context of this document) across the same or different data sources.
Data set	A data set is a collection of data that could correspond to one or more database tables.
Database	A database is a systematic collection of data that could be stored in multiple tables with data linkage between tables.
DPS core	This is the collection of core Databricks platforms, infrastructures and services offered within DPS. Among these are handling requests for adding and updating databases onto DPS.
HES_ID	A unique identifier used within HES and now replaced by *Person_ID*.
MPS_ID	Identifier returned by MPS containing the unique identifier to records in the MPS record bucket.
MPS_ID matching	The step within MPS executed after the other three tracing steps. This process will query the MPS Record Bucket to identify an existing record within this data set that matches the queried record.
MPS record bucket	Unmatched bucket of all records that have an *MPS_ID* but could not be identified as records with an *NHS number* in PDS.
NHS number	The *NHS number* is a unique identifier for a patient within the NHS in England and Wales. The specific data fields for *NHS number*s might have different names in different data sets (for example, in HES it is called *NEW_NHS_NO*). See NHS number.
Person_ID	The *Person_ID* can be either the *NHS number,* if a match for the record was found in PDS, or the *MPS_ID* if the record was matched in the MPS record bucket or sufficient demographic information was provided to create a new *MPS_ID*. If neither an *NHS number* nor an *MPS_ID* were provided, the record is assigned a *one-time-use* unique identifier as *Person_ID*.
Record	A record is an entry in the request file or response file. It can be a header or a data record. It is characterized by multiple fields.

Request file	This is the file containing all the required fields to run the MPS algorithms. This file is sent via MESH to Spine. It is composed of one header record and many data records. The data records have 23 fields.
Response file	This is the file produced by MPS (in Spine) containing all the output fields. This file is sent via MESH back to DPS. It is composed by one header record and many data records. The data records have 34 fields.
Scoring	Each PDS candidate is scored based on the similarity of features from the query record. In HES, the score is calculated from the average similarity scores of *DOB*, *postcode* and *gender*. Other data sets that contain name information also use given and family names as scoring features.
Sensitive flag	Where a patient record is flagged as sensitive, all demographic data is returned in the response; however, pipeline authors should ensure the location fields of such records are appropriately handled before data is shared with end users.
Spine	Spine supports the IT infrastructure for health and social care in England, joining together healthcare IT systems. See Spine.
Tokenisation	Tokenization (also referred to as "de-id") is the service that allows for data items to be anonymised. Currently, this happens for *NHS number*s, *Person_ID*s and *local patient identifiers*. See de-id.
Token_Person_ID	An anonymised version of *Person_ID* that acts as an identifier which allows users to count people or perform linked analysis without any need to access identifiable information. This is what most end users will use instead of the *Person_ID*.

Table of Figures and Tables

Figures

Tables

1. Introduction

1.1 Person_ID

Linking information about the same person within a data set and across data sets is essential for many analytical uses. Different data sets can include different identifying fields, data submitters may record information differently about the same person and submitted data can include data quality errors.

To verify the submitted person details and maximise opportunities for linkage between data sets, NHS England has developed the *Person_ID*, a key that identifies individuals. This uses a system known as the Master Person Service (MPS) to trace and verify key identifiers against the Personal Demographics Service (PDS). MPS can match a person's details where identifiers may be partially missing or slightly different, allowing the assignment of a consistent *Person_ID*.

NHS England data sets that assign a *Person_ID* to each record during processing include:

- Hospital Episode Statistics (HES)
- Mental Health Services Data Set (MHSDS)
- Maternity Services Data Set (MSDS)
- Community Services Data Set (CSDS)

The *Person_ID* can be either the *NHS number* if a match for the record was found in PDS, or the *MPS_ID* (a unique identifier from the MPS record bucket) if the record was not found in PDS but sufficient demographic information was provided to create a persistent unmatched record. Future occurrences of such records may be matched against the MPS record bucket if it is again not possible to find a matching record in PDS. *MPS_ID* is also sometimes known as *UPRI 1* (unmatched person record identifier 1). If the queried record could return neither an *NHS number*, nor an *MPS_ID*, then a one-time-use identifier is generated, sometimes known as *UPRI* 2 (unmatched person record identifier 2).

The *NHS number* and *MPS_ID* might be recurring identifiers across data sets, and therefore can be used to link patients.

A consistent *Person_ID* across data sets can be tokenised allowing users to count people or perform linked analysis without any need to access identifiable information. The masked identifier is sometimes known as the *Token_Person_ID*. Most users will work with the *Token_Person_ID* rather than the *Person_ID*. Specialist users from the Trusted Research Environment (TRE) can only see this tokenised identifier.

Note that this field can assume different names across different agreements (for example, *Token_Person_ID, PERSON_ID_DEID*), even if it represents the same concept. The same patient is tokenised with different identifiers depending on the data sharing agreement domain (for more information on this, please see chapter 3.10), which means that a user would not be able to link the same patient across different domains. This is done for data security purposes.

1.2 Objective and scope of this document

The aim of this handbook is to provide users of the *Person_ID* in the HES database with reference documentation on what the *Person_ID* is, how it is derived via the MPS, how the data flows between services (Data Processing Services (DPS) and Spine), and how to interpret the output information associated to the linked *Person_ID*.

This handbook contains some technical details of the MPS so users can understand how records are matched to PDS, however, it is not intended to be a complete technical guide for MPS, nor to provide recommendations on how to improve the quality of the linked records.

When a person's identity is traced by the MPS, additional information from PDS such as registered GP practice can be stored to enrich a data set. This use of the MPS is outside the scope of this document – information on the use of this functionality for a specific data set may be found in the specification for the relevant data set.

MPS is used in other settings. While this handbook may include information relevant to other uses of MPS, it should not be relied upon as a complete or accurate representation of use of the MPS in other contexts.

1.3 Questions about this document

Please direct questions to nhsdigital.personidquestions@nhs.net.

This email address should be used to give feedback about this document or to ask for clarification about any of the content of this document.

Please note, we cannot investigate specific anomalies in data returned by MPS. However, chapter 4.2 contains some specific examples of edge cases, which may be helpful.

2. Context for MPS

2.1 MPS

The Master Person Service (MPS) is operated by the Spine team. It takes the demographic information contained in a person's health and care records and matches it to their unique *NHS number* held in PDS to confirm their identity.

The matching or linkage process is comprised of several steps of increasing computational intensity, designed to address the challenges provided by data sets of different degrees of quality.

The outcome of the search might be an *NHS number* if the record was matched in PDS, an *MPS_ID* if the record was matched with previously unmatched records. If neither search was successful, a *one-time-use ID* is generated in DPS. DPS then transforms these into the *Person_ID* field.

MPS is exclusively used by DPS; data sets are batch-processed and enriched of the *Person_ID* field and possibly other fields (depending on the needs of the data asset).

There are other services in NHS England that support the retrieval of patients' *NHS number* and other demographic information (for example, live services used in direct care). These, however, do not use MPS, but separate services which rely on similar tracing algorithms.

2.2 PDS

MPS uses the Personal Demographics Service data set (PDS); PDS is the national electronic database of NHS patient details, which holds about 80M records. The demographic details are normally updated when patients visit their GP but may also be updated by other healthcare professionals. This is therefore a "live" database as it changes daily. Data on PDS helps care providers to confirm the identity of patients, link their care records within an organisation and between different organisations, and to communicate with patients.

MPS uses PDS as the source database against which other data sets are compared for a match. In other contexts, PDS is sometimes referred to as a "service", but in the context of MPS (and this document), PDS is referred to as a "database".

The data items held include *NHS number*, *name*, *date of birth* (*DOB*), *gender*, *GP practice*, *addresses* (including historic addresses and postcodes) and *contact details* (such as telephone numbers and email addresses). Data is also held, where applicable, on certain

patient preferences such as nominated pharmacy and whether the record is marked as 'sensitive'. No clinical data is held on the PDS. The full list of data items held on the PDS, along with other information about PDS, can be found on the Demographics pages of the NHS Digital website.

The National Back Office (NBO) provides a national data quality service, responsible for resolving incidents with patients' demographic records in the PDS. NBO services include the identification of duplicates, confusions and changes of identity (for example, in adoptions and gender reassignment cases where a new *NHS number* has been assigned to the same person). The Demographics and NBO teams work closely together on investigations of more complex incidents and issues.

2.3 Person_ID vs. NHS number

*NHS number*s are a national patient identifier within the UK's health and social care system. The official list of individuals' *NHS numbers* is held within PDS. The *NHS number*s are allocated to newborn babies soon after birth, and to individuals that are registered for NHS care in England, Wales or the Isle of Man. However, some individuals do not have one, for example, overseas visitors, long-term mental health patients and private patients.

The *Person_ID* derived via MPS uses the *NHS number* when this is available but creates an alternative key when the details of an individual are not found within PDS.

2.4 Person_ID and the use of MPS on HES

Hospital Episode Statistics (HES) consists of three databases containing all admitted patient care (APC) admissions, outpatient appointments (OP) and accidents and emergency (A&E) attendances at hospitals in England.

This is a well curated data asset, and most records in HES data sets have *NHS number* and *DOB* populated. This means that *Person_ID* matching is mostly based on the first, simplest and most robust tracing step in MPS, that is, cross-check trace.

HES data sets do not include patient names (surnames or forenames). As a result, the only other tracing step that can be used on HES is the algorithmic trace as the alphanumeric trace step must use names.

Data sets may have had their own methods of assigning a person identifier, and custom linkages based on combinations of data items may have been in use prior to data sets adopting

the *Person_ID*. In HES, for example, *Person_ID* has replaced the use of *HES_ID*. Individual assessments of impacts of the adoption of *Person_ID* are undertaken on a case-by-case basis and published on the Methodological changes page (see “Announcement of methodological change to HES”).

Notably, HES runs through MPS after each submission and refreshes the *Person_ID* with the most up to date data available.

3. Technical details of the data linkage algorithm

3.1 Overview of MPS

The process to link a given record to an existing person identifier happens across DPS and Spine as described in Figure 1.

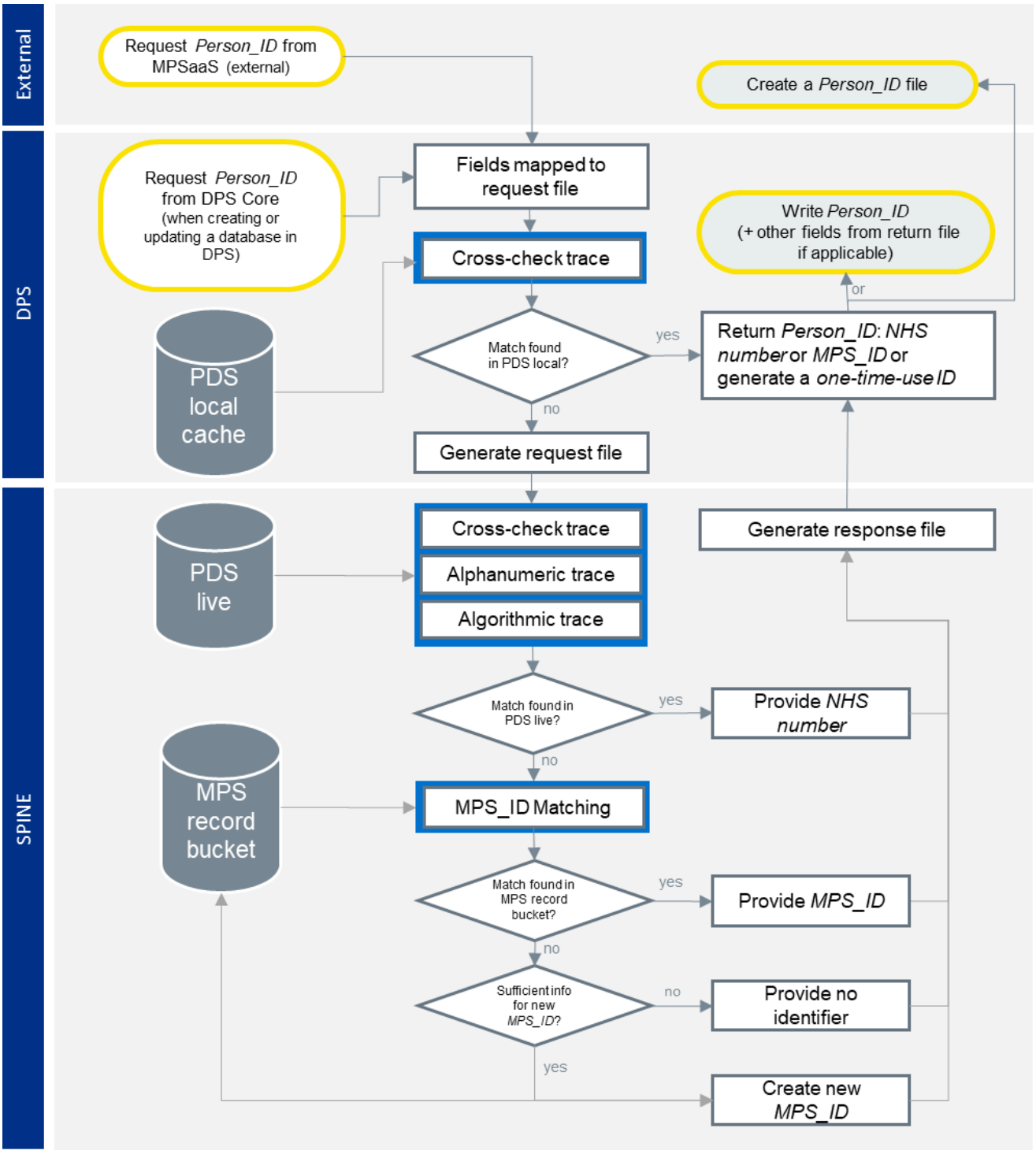

Figure 1. Creation of *Person_ID* via MPS process flow (high-level)

Figure 1 illustrates the entire workflow of how the *Person_ID* is assigned to a data set. This section provides a high-level explanation of the diagram, while more details are given in the following sections.

The workflow starts on DPS core with the request of the *Person_ID* that corresponds to a record in the data set, this is also referred to as the 'request query record'. The request query can also be triggered outside of DPS as part of Master Person Service as a Service (MPSaaS) which is out of the scope of this document.

The pipeline in DPS core first maps the data to the request file fields. Where the *NHS number* and *DOB* are provided, DPS cross-checks them against the local cached version of PDS and confirms the identity. This is the most straightforward step.

The records that could not be matched at this stage (either because one of these fields is empty or the information does not match with the reference, that is, PDS) are collected into a request file and sent via the Message Exchange for Social Care and Health (MESH) to MPS in Spine.

PDS is updated daily on DPS core, which makes the PDS cached version slightly different from the live version of PDS in Spine.

The request file goes through quality checks in MPS, and the demographic fields available for the record are used for the following tracing steps against the live version of PDS in Spine.

There are three tracing steps, cross-check, alphanumeric and algorithmic, the latter is the most computationally expensive. The tracing against PDS stops whenever a successful match is found, or if the record does not meet the eligibility criteria to proceed to the following step, or at the end of the last step. The HES databases skip the alphanumeric trace because the fields needed for this step are not available. For completeness, this is still discussed in the present document.

On top of *NHS number* and *DOB*, the cross-check trace in Spine uses *name* and *outbound postcode* if available. If no perfect match is found, the algorithm proceeds to the alphanumeric trace, where the mandatory fields (*family name*, *year of birth* and *gender*) are used to identify a match on PDS. If a match is not found (or the trace is not run for lack of demographics), MPS proceeds to the algorithmic trace step, where the single query record is compared to all records in PDS. The comparisons involve the same demographic information mentioned above plus *gender* and *full postcode*, and are scored based on similarity. If the similarity is deemed acceptable, the matched record is returned. Otherwise, the algorithm proceeds to look for similarities to previously unmatched records, stored in the MPS record bucket, a separate data set.

The MPS record bucket in the Spine's data store is used to link records for the same person that cannot be traced in PDS. If a match is found, an *MPS_ID* is returned, otherwise, the algorithm considers whether to create a new *MPS_ID*. *MPS_ID* is also known as *UPRI* 1 in certain documentations.

A new *MPS_ID* can be created only when the minimum required fields are provided. If this is not the case, an empty *MPS_ID* field is returned, and DPS core generates a *one-time-use ID* (also known as *UPRI* 2).

The MPS users will submit a file (request file) with the records that need to be matched. The file must follow the technical specification (see Appendix 5.1) which describes the 25 fields used in the matching process.

Once all the records have been processed in Spine, MPS will return a file (response file, see Appendix 5.2) with the details of the transaction, like the number of data records included in the file, and the fields of interest such as the *NHS number*, the *MPS_ID*, and others that provide information on the matching process.

3.2 Input data and validation

HES data sets obtain the *Person_ID* field when processed via DPS core pipelines, similarly to any other data set in DPS which requires MPS tracing.

Among other things, it is the responsibility of the DPS pipeline author to:

- map the submitted fields into the MPS request schema – the request file is composed of the fields described in Table 9 (Appendix 5.1). Not all the fields are required for MPS to find a match
- make sure that records are deduplicated based on the available demographic fields – this is done to reduce the computational burden

This may require varying levels of preparation work depending on whether all the relevant information is available within the same table, or whether it needs to be retrieved from multiple locations/tables.

The creation of the request file happens immediately after the data set has been processed via cross-check trace in DPS. Only the records that could not be matched in DPS are sent to MPS in Spine for further processing.

Before proceeding to the other tracing steps, the fields in the request file are validated with basic checks looking at whether mandatory data are present, data type, format and strings

length are consistent. If the request file fails the validation, the entire process is unsuccessful, and the data is not processed further.

Once the data has been validated, empty fields are removed. The family, given and any other name are all changed to upper case. Invalid characters[1] are removed from all fields except *local patient identifier, internal identifier, telephone number, mobile number* and *e-mail address*. The expected format of the *postcode* field is with a space in the middle. Such space is then replaced with underscore to match the postcodes in PDS. If this is not the case, the *postcode* field might not match correctly.

3.3 Cross-check trace (DPS and Spine steps)

Cross-check trace is the first and simplest tracing step, which can be used when an *NHS number* is present. The vast majority (above 99%) of HES records are matched using cross-check trace (data referring to the most recent HES tables at the time of writing, that is, 2021/2022).

As illustrated in Figure 1, cross-check trace is repeated twice, once in DPS against the cached version of PDS, and another time in Spine against the live version of PDS.

The cross-check trace in DPS looks for an exact match of the provided *NHS number* and *DOB* (Figure 2).

[1] This is the list of invalid characters: '!', '$', '%', '&', '(', ')', '[', ']', '{', '}', '=', ':', ';', 'Number', '~', '@', '|', '<', '>', '.', '?', '/', '_', '\\', '\xc2\xa3'

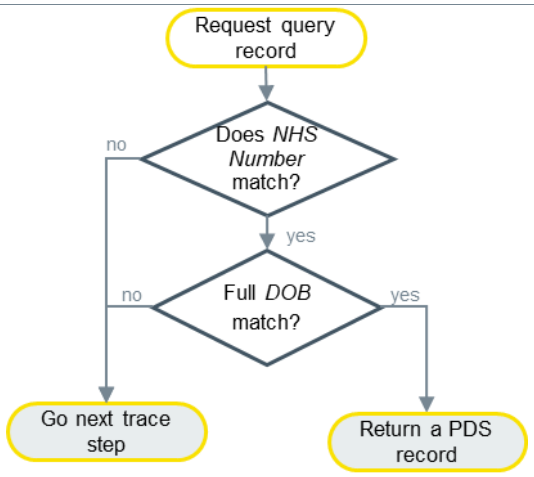

Figure 2. Cross-check trace process flow in DPS core

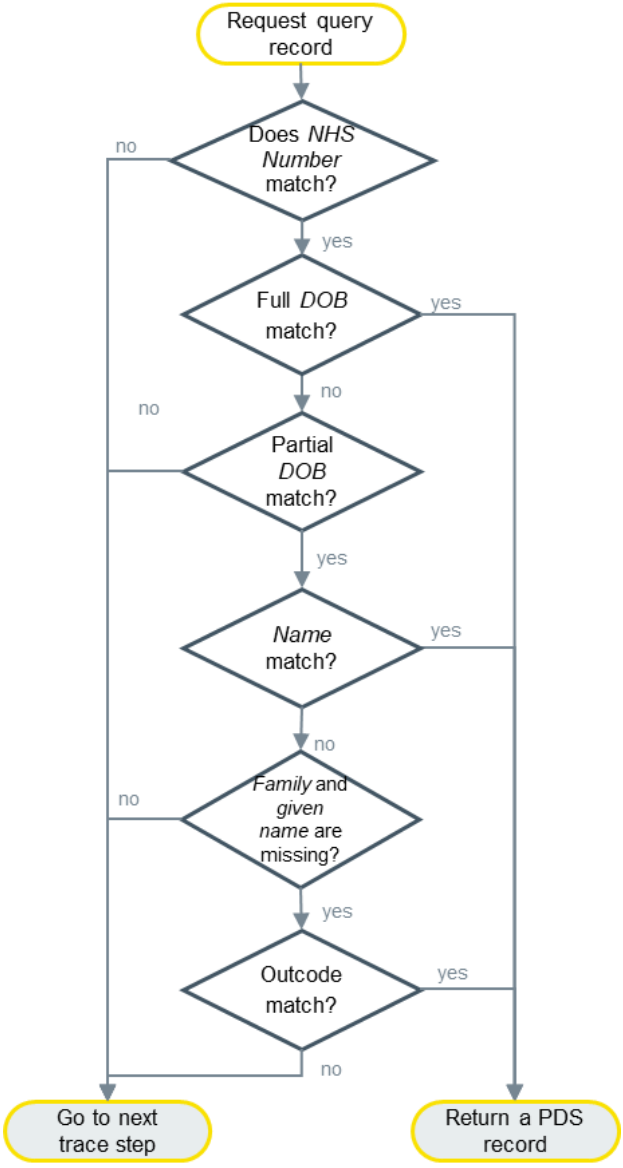

Figure 3. Cross-check trace process flow in Spine

The diagram in Figure 3 describes the cross-check trace algorithm carried out in Spine. Besides *NHS number* and *DOB*, it also includes a partial *DOB* check and checks on the name and outcode values, that is, the left part of the postcode before the single space in the middle of the postcode.

Perfect matches for *NHS number* and *DOB* immediately return a PDS match. If instead the *DOB* only matches partially, then it checks name or outcode values.

Partial *DOB* match

A partial *DOB* match is where at least 2 of day (DD), month (MM), and year (YYYY) match, allowing for:

- DD/MM being swapped to MM/DD, for example, 12/06 becomes 06/12
- D_1D_2 being swapped to D_2D_1, for example, 12 becomes 21
- $Y_1Y_2Y_3Y_4$ being swapped to $Y_1Y_2Y_4Y_3$, for example, 1945 becomes 1954

Name match

A name match is where:

- *given name* is present and matches on the first character, and
- *family name* is present and matches on the first three characters

If *given name* or *family name* are not present, then the algorithm tries an outcode match instead. In HES, names are never provided, so this check automatically fails.

Outcode match

The outcode match is where the first part of a postcode (for example LS17) of the query matches the first part of the current or any of the historic postcodes on the PDS record.

Scoring

If a PDS record is returned at this stage (both in the DPS and in the Spine cross-check trace steps), it is considered certain and it gets a score of 100 in the *MatchedConfidencePercentage* field in the response file.

Superseded NHS numbers

An *NHS number* can be superseded in PDS, which means that it is no longer valid, and it has been replaced by another one. If a query record contains a superseded *NHS number*, cross-check trace in DPS does not recognize this as a match and the record is processed via cross-check trace in Spine. Spine cross-check trace is capable of recognizing such matches, but returns the corresponding valid *NHS number* as a matched record rather than the submitted superseded one.

3.4 Alphanumeric trace

Alphanumeric trace is the stage after cross-check trace and before algorithmic trace. Records only reach alphanumeric trace when no match is found in the previous trace step, and if the query record contains the following mandatory fields:

- *family name*
- *year of birth*
- *gender*

In addition, alphanumeric uses the following features as non-mandatory fields:

- *DOB* (full)
- *given name*
- *postcode*
- *GP provider*
- *date of death*
- *Royal Mail's Postcode Address File*[2] *(PAF) address*

[2] https://www.poweredbypaf.com/

The logic of alphanumeric trace is described in Figure 4.

First, the record is checked to see if it contains all the mandatory fields. Failing this, it goes to the next trace step. For query records with all the mandatory fields, alphanumeric trace will filter candidate records in PDS using the following fields if they exist in the request query record:

- current *family name* (exact Soundex match)
- current *gender* (exact match)
- current and historical *DOB* (exact match)
    - if the query record has a partial *DOB* then an exact match on the partial *DOB* is required
- historical or current *PAF address* (exact match)
- historical or current *GP Provider* (exact match)
- historical or current *postcode* (exact match)
- historical or current *given name* (exact Soundex match)
- *date of death* (exact match)

Soundex calculation is described in Section 5.4. If only one candidate record from PDS is found at the end of the filtering steps, then this is returned as matched record. If no candidate is left or more than one candidate is found, then the request query record goes to the next trace step.

Records with a partial Date of Deaths

If a query record contains a partial *date of death* (for example, just the year of death) then alphanumeric trace will skip the checking of the minimum mandatory fields.

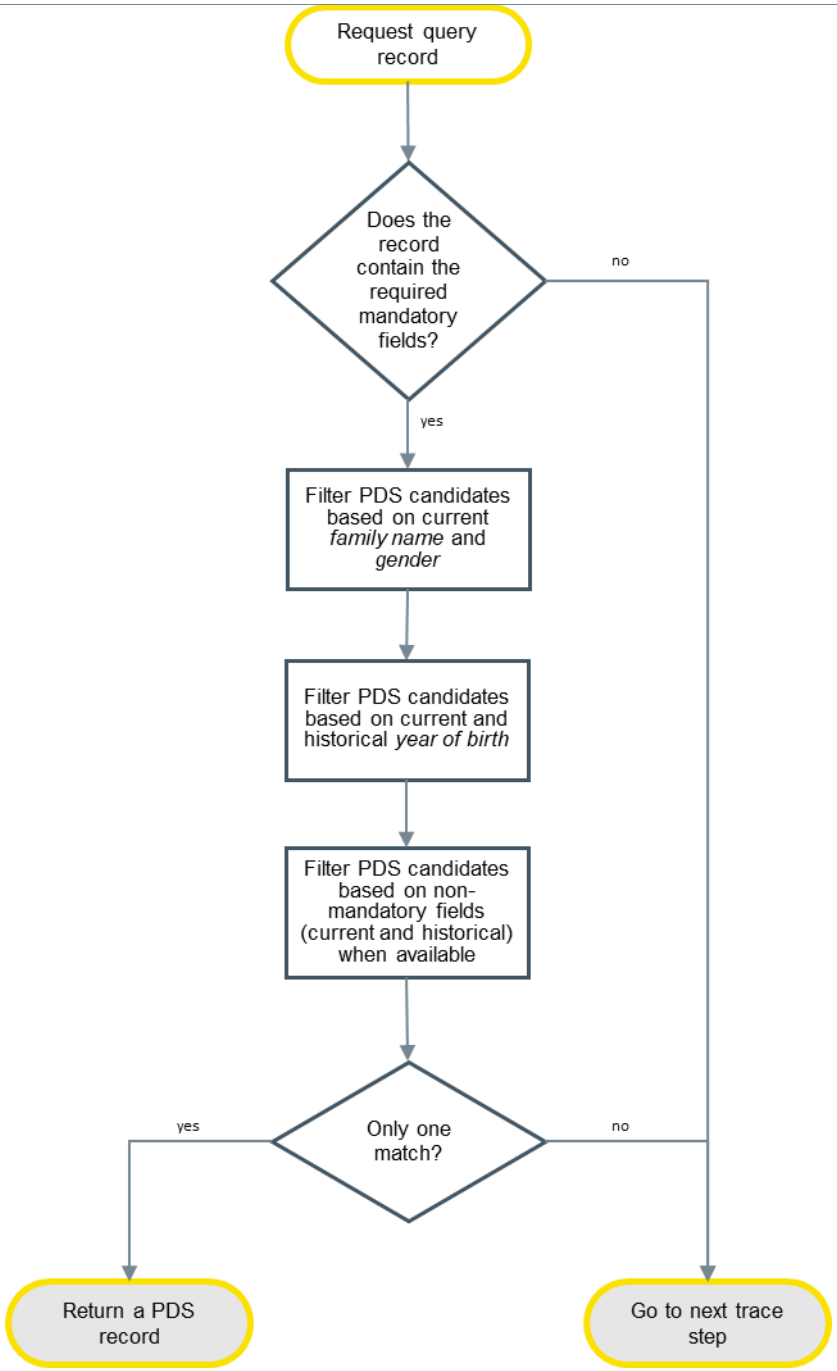

Figure 4. Alphanumeric trace process flow in Spine

Examples of alphanumeric trace

The following examples show common outputs from alphanumeric trace step. First, consider PDS only consisting of the following records:

	Mandatory Field			Non-mandatory Field				
Id	**Family Name**	**DOB**	**Gender**	**Given Name**	**Postcode**	**GP Provider**	**Date of Death**	**Royal Mail Address**
1	Bernard	1992-01-01	1	Sammy	SW1A 2AB	000001		
2	Cherry	1976-08-15	2	Penelope	E14 5EA	000002		
3	Fox	2002-12-17	1	Hadley	SE1 8UG (start to 2005-01-02); LS1 4AP (2005-01-02 to present)	000003		

FICTICIOUS DATA

Example 1: Consider the following query record where the mandatory field *gender* is missing.

Family Name	DOB	Gender	Given Name	Postcode	GP Provider	Date of Death	Royal Mail Address
Bernard	1992-01-01		Sammy	SW1A 2AB			

FICTICIOUS DATA

The query record will move to the next trace step due to missing *gender* in the mandatory field.

Example 2: Consider the following query record where non-mandatory fields are missing.

Family Name	DOB	Gender	Given Name	Postcode	GP Provider	Date of Death	Royal Mail Address
Cherry	1976-08-15	2	Penelope	E14 5EA			

FICTICIOUS DATA

The query record will match with record id 2 in PDS even though the *GP Provider* is null in the query field.

Example 3: Consider the following query record where one of the non-mandatory fields is different from the PDS record.

Family Name	DOB	Gender	Given Name	Postcode	GP Provider	Date of Death	Royal Mail Address
Fox	2002-12-17	M	Hadley	LS1 4AP	000009		

FICTICIOUS DATA

The query file will move to the next step as there is a mismatch in the *GP Provider.*

Example 4: Consider the following query record where one of the non-mandatory fields matches on historical values

Family Name	DOB	Gender	Given Name	Postcode	GP Provider	Date of Death	Royal Mail Address
Fox	2002-12-17	M	Hadley	SE1 8UG			

FICTICIOUS DATA

The query file will match with id 3 in PDS, the postcode matches on a historical value in PDS.

3.5 Algorithmic trace

Algorithmic trace is the final stage in MPS to match records with PDS, and it is run if no match was found with cross-check trace or alphanumeric trace. The minimum eligibility criteria for running this step are having valid values in the *DOB*, *gender* and *postcode* fields. An overview of the workflow of the algorithmic trace is described in Figure 5.

Algorithmic trace can be summarised as follows: for each query record, a set of PDS candidate records are identified by blocking on some demographic fields; 50 or fewer records are blocked (filtered) and scored. The highest scoring candidate record is chosen as the matching record, and it is returned. However, if no candidates are found, or the highest scoring candidate cannot be resolved (for example there are multiple close matches) then no PDS record is returned.

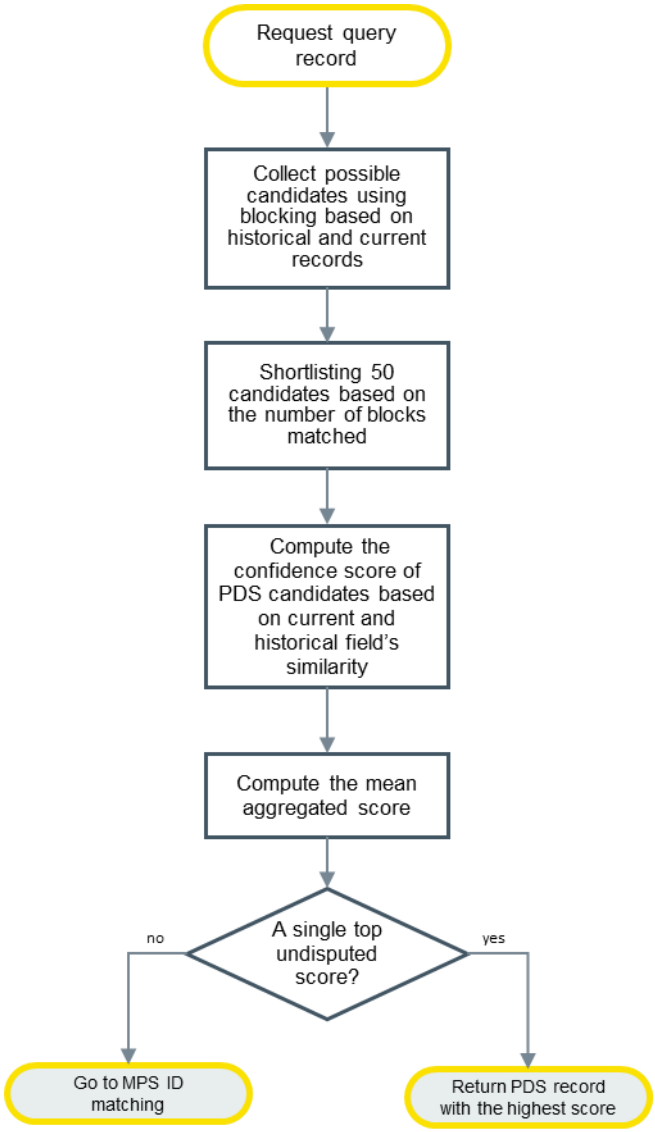

Figure 5: Algorithmic trace for all records including HES

Blocking

The key principle of the algorithmic trace is block matching or blocking, where PDS records are considered candidates for the query record if they match to a block of demographic characteristics.

Before the blocking stage takes place, *family name* and *given names* are pre-processed by removing spaces and hyphens and then mapping via a dictionary into a normalised name[3]. The algorithmic traces consider the following blocks:

- Soundex of *family name,* Soundex of *given name*, *DOB*
- Soundex of *family name, gender, DOB*, *postcode*
- Soundex of *given name*, *gender*, *DOB*, *postcode*
- *DOB*, *postcode*, *gender*

Soundex calculation is described in Section 5.4. The PDS records need an exact match on the elements of the block to be considered as a candidate. However, the exact match is accepted on either the current or the historical values of the features except for *gender* which must match exactly on current value including unknown and indeterminate values.

A maximum of 50 candidates are retained from the blocking step, and these are chosen starting from the PDS records that matched on the highest number of blocks (that is, PDS records that matched on all 4 blocks are included first in the list of candidates). For HES records where names are unavailable, algorithmic trace can only use the last blocking rule based on *DOB*, *postcode* and *gender*.

Scoring & Ranking algorithmic trace on HES

Notably, in HES we do not have features with partial scores. This is because the only block available in algorithmic trace is block 4 (*DOB*, *postcode*, *gender*), and PDS records need perfect matches on all features to be considered candidates. Hence, all PDS candidate records in algorithmic trace will have a perfect match and therefore an algorithmic score of 100.

This also makes the ranking inconsequential because all the selected candidates have equal perfect scores. Hence, where multiple candidates are found, they are all rejected with error code 97 (that is, multiple matches found). When only one candidate is identified by blocking, algorithmic trace can successfully return a match.

The following sections on scoring and ranking are provided for guidance on the operation of MPS for other datasets.

[3] The dictionary maps input names like “Jenny” to the full form “Jennifer”. The full content of the dictionary is available in the supplementary material.

Scoring

Each PDS candidate is scored based on the similarity of features from the query record. The score is calculated from the average of similarity scores of *DOB*, *postcode*, *gender* and the name instance(not available for HES). The name instance is a combination of *given name*, *other given name* and *family name* at a specific point in time. The scoring uses the original entries in PDS and not the normalised version from the blocking stage. If any of the features are missing or null, they are not included in the average calculation.

- For *DOB*, the YYYYMMDD dates are scored based on the rules in Table 1.

Table 1. Algorithmic trace scoring system for *DOB*

Condition	Score
Match on YYYYMMDD	100
Match on MM and DD only	66
Match on YYYY and MM only	66
Match on YYYY and DD only	66
Match on YYYY and MMDD transposed matches	66
Match on YYYY only	33
All other states	0

- For *gender*, the scores are based on the rules in Table 2.

Table 2. Algorithmic trace scoring system for *gender*

PDS gender	Query gender			
	Not known	Male	Female	Not specified
Not known	100	50	50	50
Male	50	100	0	50
Female	50	0	100	50
Not specified	50	50	50	100

- For *postcode*, the scoring is summarised in Figure 6.

The scoring considers the current and historical home address postcodes separately. It starts with the *current postcode*, and there are three possible outcomes: an exact, partial or no match. To achieve an exact match, all *postcode* characters between PDS and query must match. This produces a score of 100.

A partial score can be obtained if the query record contains a partial postcode, and these *n* characters match perfectly the first *n* characters of the postcodes in the PDS candidate records. In this case, the score would be the length of the *postcode* in the query record divided by the length of the *postcode* in the candidate record. For example, the *postcode* in the query record is LS1 and the *postcode* on the PDS record is LS1 4AP, the length of the *postcode* in the query record is 3. Because it matches the first 3 characters of the PDS record *postcode*, then the postcode score is 3/7*100 = 43 (approximation to no decimal points).

If none of the *current postcodes* (usually there is only one current home address postcode) have scored above zero, the algorithm considers *historical postcodes* on the PDS candidate records. All *historical postcodes* are scored with either exact or partial scores, and the highest score is taken as a match.

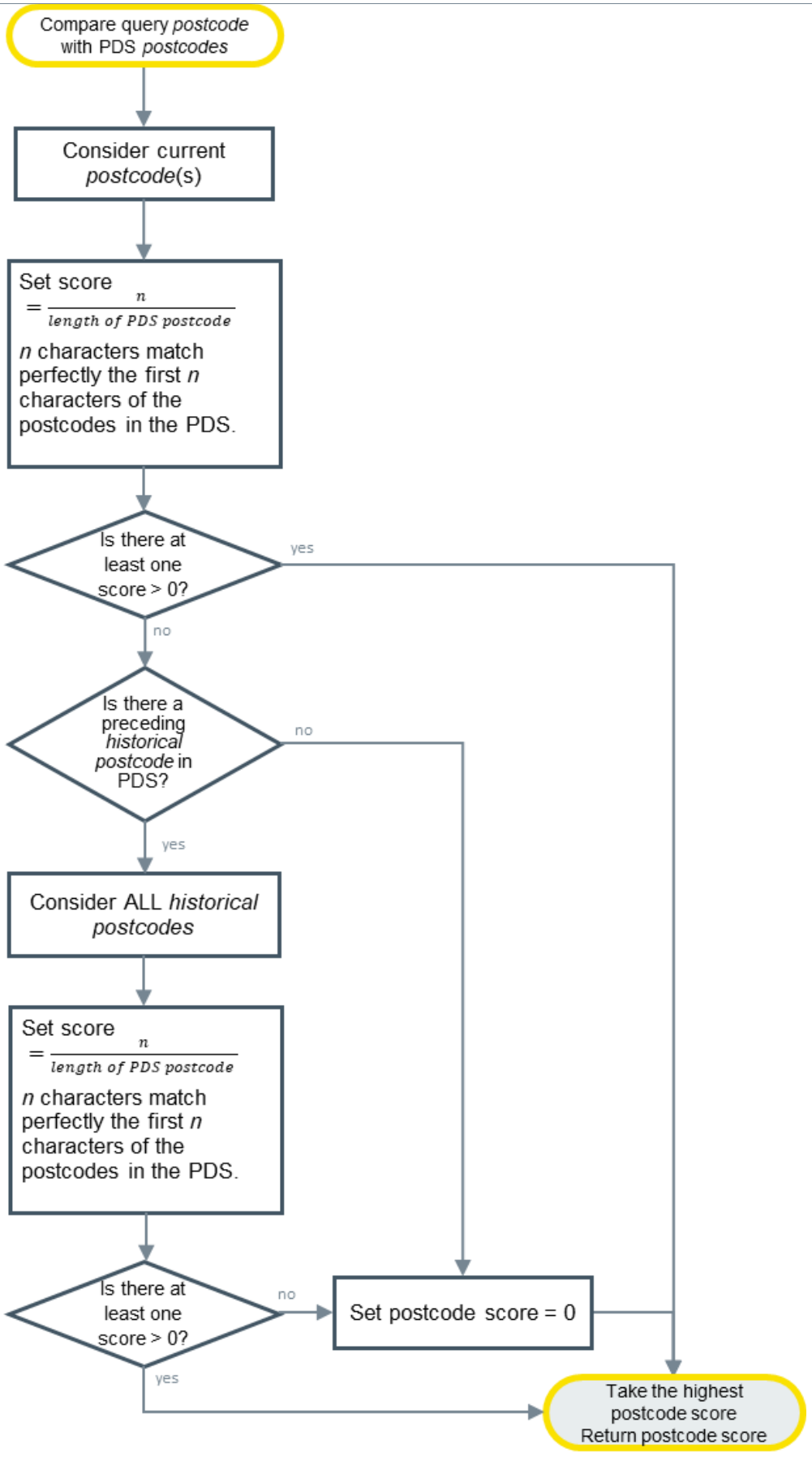

Figure 6. Compute postcode score in algorithmic trace process flow

- For the name instance, the scoring is summarised in Figure 7.

The name instance is a set of *given name, other given name* and *family name* values at the same point in time. Notably, *other given name* is only included in the matching if the field is non-null on the query record. For each current and historical name instance, any non-ASCII characters are converted to a "@" character. The Jaro-Winkler score is then computed. The highest total score over all name instances is returned.

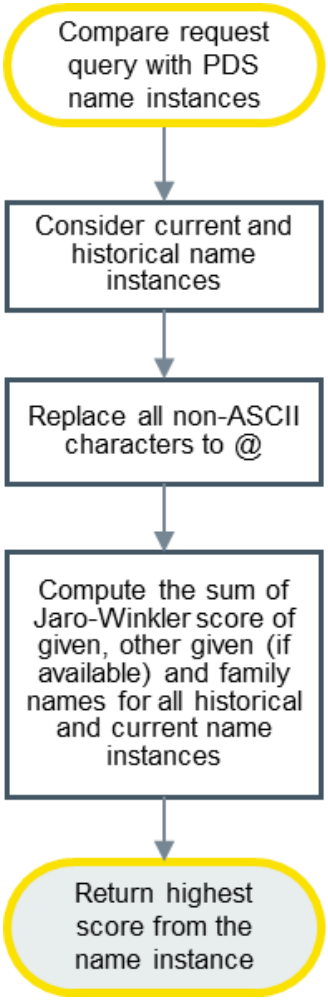

Figure 7. Compute name score in algorithmic trace process flow

Family Name and Given Name Confidence Score in Response Record

The MPS response file contains distinct confidence percentage scores for *given name* and *family name*. However, the user should be aware that these might differ from the scores calculated for the name instance as explained above. The ranking step in algorithmic trace uses the name instance score, and not the confidence percentage scores for *given name* and *family name* provided in the response file and consequently in the HES fields.

Moreover, if the query record has non-null *other given name* then the highest Jaro-Winkler score between *given name* and *other given name* will be taken as the *given name* confidence percentage score in the response record.

The users might encounter occasional inconsistencies in the algorithmic trace scores that can be linked back to this explanation.

Ranking

For each combination of PDS candidate and query records, a score (between 0 and 100) is calculated as the average of the similarity scores across all non-null features. For HES, two of the five fields (*given name* and *family name*) are always null, and the remaining three (*DOB*, *postcode* and *gender*) are required to be non-null for algorithmic trace to be carried out. So, for HES the score will be an average over similarity scores for three fields.

Candidates from all blocks are ranked and algorithmic trace returns the matched PDS record with the highest score. If two or more of the highest-ranking PDS candidates have similar scores (within 5 points), then algorithmic trace does not return a match. For example, if the highest-ranking PDS candidate achieved a score of 95 and the second highest of 91, they are both rejected because they are too close to disambiguate, and the record will be returned with an error code of 97 (see chapter 5.3).

Examples for algorithmic trace

The following examples show common scenarios in the blocking and scoring stage of the algorithmic trace step.

Example 1: Preprocessing of names in blocking

Consider the following query record at the start of the blocking step.

Given Name	Other Given Name	Family Name	Date of Birth	Gender	Postcode
Jon		Jones-Smith	1992-01-01	1	SW1A 2AA

FICTICIOUS DATA

In the preprocessing the following happens:

- Jon becomes Johnny because of the name normalisation
- The hyphen is removed from Jones-Smith which become Jonessmith

Thus, the fields used for the blocking are

Given Name	Other Given Name	Family Name	Date of Birth	Gender	Postcode
Johnny		Jonessmith	1992-01-01	1	SW1A 2AA

FICTICIOUS DATA

Example 2: Scoring of names with hyphens

Consider the comparison of the following query record with one candidate from the blocking stage:

	Given Name	Other Given Name	Family Name	Date of Birth	Gender	Postcode
Query Record	Jon		Smith-Jones	1992-01-01	1	SW1A 2AA
Candidate	James		Smith	1992-01-01	1	SW1A 2AA

FICTICIOUS DATA

The scoring is calculated as the following:

- For *given name*, the Jaro-Winkler distance is computed between “Jon” and “James” which gives 51
- For *family name*, hyphens is an ASCII character and are not replaced with @. Thus, Jaro Winkler is computed between “Smith-Jones” and “Smith” which gives 89
- There is an exact match on all other fields thus scoring 100

	Given Name	Other Given Name	Family Name	Date of Birth	Gender	Postcode
Candidate 1	51		89	100	100	100

The final aggregated score is the mean average of all 5 fields, that is 88 in this case.

Example 3: Scoring of candidates with non-ASCII signs

Consider the following query records providing 2 candidates from the blocking stage:

	Given Name	Other Given Name	Family Name	Date of Birth	Gender	Postcode
Query Record	Zöe		Ó Briain	1992-01-01	2	SW1A 2AA
Candidate 1	Zöe		O Briain	1992-01-01	2	SW1A 2AA
Candidate 2	Zoe		Briain	1992-01-01	2	SW1A 2AA

FICTICIOUS DATA

- For *given name*, the scores are calculated as follows:
    - For Candidate 1, the name “Zöe” is transformed to “Z@e”. Thus, the Jaro-Winkler score is computed between “Z@e” from query record and “Z@e” from Candidate 1, which gives 100
    - For Candidate 2, the Jaro-Winkler score is computed between “Z@e” from the query record and “Zoe” from Candidate 2, which gives 80
- For *family name*, the scores are calculated as follows:
    - For Candidate 1, the name “Ó Briain” is transformed to “@ Briain”. Thus, the Jaro-Winkler score is computed by comparing “@ Briain” with “O Briain”, which gives 92
    - For Candidate 2, the Jaro-Winkler score compares “@ Briain” with “Briain”, which gives 80
- All other fields have a perfect match thus scoring 100

	Given Name	Other Given Name	Family Name	Date of Birth	Gender	Postcode
Candidate 1	100		92	100	100	100
Candidate 2	80		92	100	100	100

The final aggerated score is the mean average of all 5 fields, that is 98 for Candidate 1 and 94 for Candidate 2.

No match is returned in this case, as there is a less than 5-point gap between the highest and the second highest score.

Example 4: Scoring of candidates with a non-null *other given name* field

Consider the comparison of the following query record with 3 candidates from the blocking stage:

	Given Name	Other Given Name	Family Name	Date of Birth	Gender	Postcode
Query Record	John	Adams	Smith	1992-01-01	1	SW1A 2AA
Candidate 1	Jon		Smith	1992-01-01	1	SW1A 2AA
Candidate 2	Jon	Adams	Smith	1992-01-01	1	SW1A 2AA
Candidate 3	John	Dan	Smith	1992-01-01	1	SW1A 2AA

FICTICIOUS DATA

- For *given name*, the scores are calculated as follows:
    - For candidate 1 and 2, the Jaro-Winkler score is computed comparing “John” and “Jon”, which gives 93.
    - For candidate 3, the Jaro-Winkler score is computed using “John” and “John”, which gives 100
- For *other given name*
    - For candidate 1, the field is null thus the Jaro-Winkler score is 0.
    - For candidate 2, the Jaro-Winkler score is computed using “Adams” and “Adams”, which gives 100.
    - For candidate 3, the Jaro-Winkler score is computed using “Adams” and “Dan”, which gives 51.
- For the other fields, there in an exact match for all candidates thus scoring 100.

	Given Name	Other Given Name	Family Name	Date of Birth	Gender	Postcode
Candidate 1	93	0	100	100	100	100
Candidate 2	93	100	100	100	100	100
Candidate 3	100	51	100	100	100	100

The final aggregate score is the mean average for all 6 fields, that is, 82 for candidate 1, 99 for candidate 2 and 91 for candidate 3.

In this case candidate 2 is returned as match.

3.6 MPS_ID matching

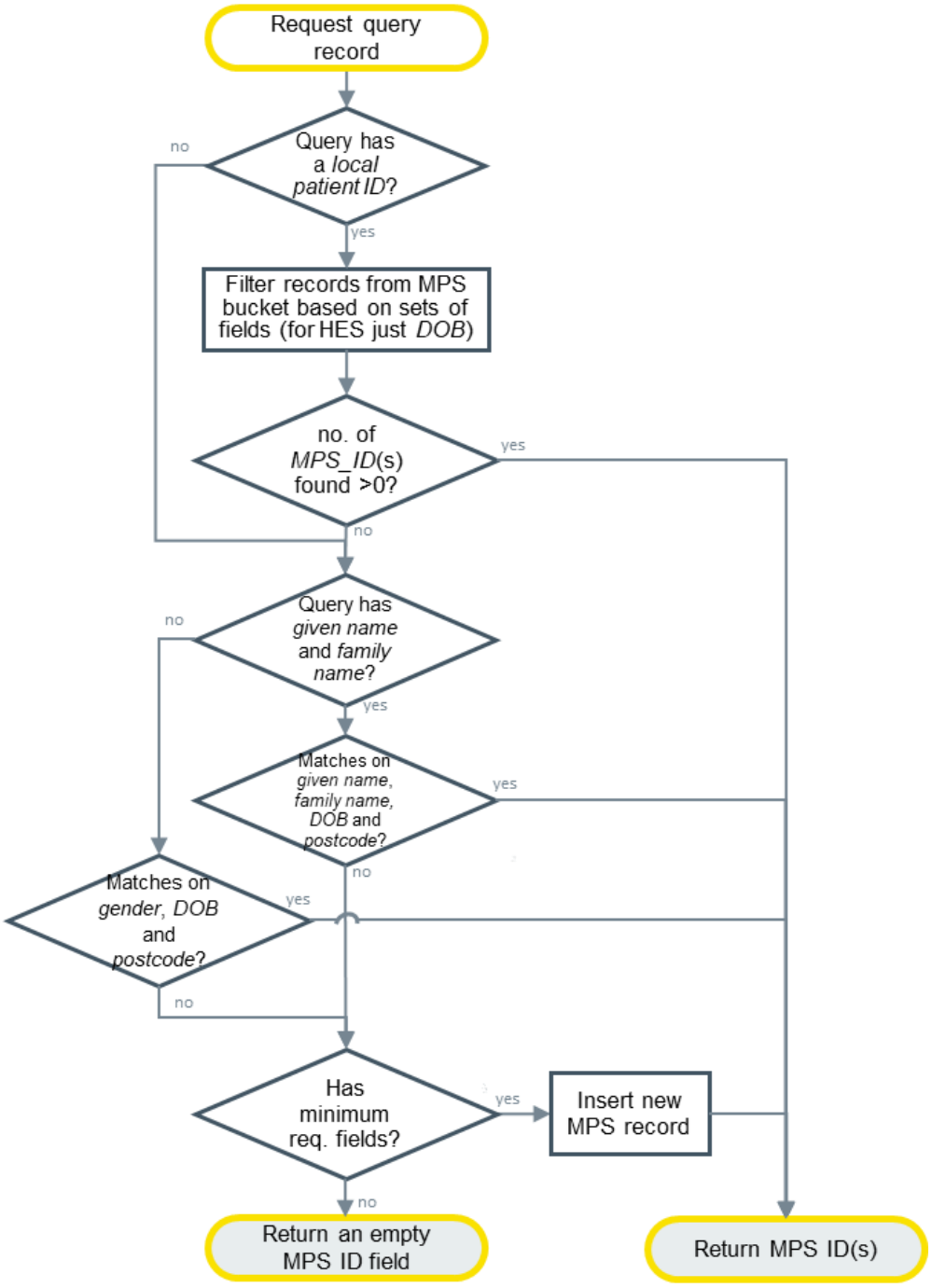

Figure 8. MPS_ID matching process flow

The last step in MPS consists of running MPS_ID matching against the MPS record bucket as illustrated in Figure 8. If the query record matches an existing record in the MPS record bucket, then the *MPS_ID* is returned. Otherwise, if the minimum required fields are provided, then a new *MPS* record is generated and stored in the MPS record bucket.

An empty *MPS_ID* field is returned if the query record does not have the minimum required fields.

Initial checks

MPS_ID matching is run only if the previous tracing steps returned an invalid *NHS number* (that is, if the *NHS number* is 0000000000).

MPS query with *local patient ID*

This step is run if the query record includes a *local patient ID*. All the MPS records with *local patient ID* corresponding to the query record are selected if any of the following sets of fields match:

- *family name*, *DOB*, *postcode* and either *given name* or *gender*
- *given name*, *DOB* and *postcode*
- *given name*, *family name*, *gender* and *postcode*
- *given name*, *family name*, and *DOB*
- *DOB* only (if *given name* or *family name* are missing) – this is the only set that is relevant for HES

MPS query without *local patient ID*

If the query does not include a *local patient ID*, or if the previous step returned no *MPS_ID*s, then this step returns all records in the MPS record bucket which match one of the following:

- if the query record of the request file contains a *given* name and *family name*, then *given name* and *family name* must both match the record in the MPS record bucket (along with *DOB* and *Postcode*)
- if the query record does not contain a *given name* or *family name*, then *gender* must match the record in the MPS record bucket (along with *DOB* and *Postcode*)

Multiple MPS_ID Matches

MPS matching can return multiple *MPS_ID* matches for the same record. This can occur because this is a rigid search and the records are not scored like in the algorithmic trace step, so all the matched records are considered equally good. We can find multiple matches if a record in the MPS bucket is duplicated in multiple records with different *MPS_ID*s. There is no current process to remove duplicates (unlike PDS, which is curated by the National Back Office), and hence multiple matches are returned in the response file.

When the HES pipeline in DPS processes the response file, multiple matches are lost because only the first identifier is picked. The first identifier will not consistently be the same one, so these records might be randomly allocated to one of the duplicates over time.

Minimum required fields

If neither of the above queries returns any *MPS_ID*s, then the following step creates a new MPS record. The minimum required fields for creating a new MPS record are valid *DOB* and either *local patient identifier* or valid *postcode*.

MPS record bucket maintenance

Records in the MPS record bucket are not updated based on new queries. For example, if a query record is matched to an *MPS_ID* but the *postcode* is different, this new *postcode* is not added to the MPS record information.

Differently than PDS, MPS record bucket is not periodically checked for duplicates or inconsistent and impossible records.

An MPS_ID can be used to link records across datasets.

Which records can populate MPS record bucket?

In the process above we described that, if a match is not found but the query record has sufficient information for the creation of a new MPS_ID record, then this is added to the MPS record bucket.

However, there is a control mechanism in place to prevent the addition of unsuitable records, such as the ones that have a retention period linked to a research study. This is the case for many cohorts that external organisations submit via DARS (Data Access Request Service)

with the purpose of linking these patients' data to other assets controlled by NHS England. These records will be assigned a *one-time-use ID* instead of a *MPS_ID*.

In summary, the cohort data transmitted through DARS does not ultimately populate the MPS record bucket.

3.7 One-time-use ID

The records that could not be matched neither with PDS, nor with MPS record bucket, and that did not have sufficient information to generate a new *MPS_ID*, are left unmatched by MPS. Such unmatched records have a one-time use ID generated in DPS, so that all the different unmatched records are kept distinct. This identifier is also known as *UPRI* 2 (unmatched person record identifier 2).

The one-time use IDs cannot be used to link records across datasets, or even within a dataset.

3.8 Scoring and other outputs

The results from MPS are provided in the response file. The response file is only used in DPS for internal purposes (for example, creating the final *Person_ID* field, as explained in chapter 3.9).

The fields of this file are detailed in Appendix 5.2, but in summary, they are the same fields as the request file with the addition of specific MPS output fields as follows:

- *SENSITIVE FLAG*
- *ERROR/SUCCESS_CODE*
- *MATCHED _NHS_NO*
- *MPS_ID*
- *MatchedAlgorithmIndicator*
- *MatchedConfidencePercentage*
- *FamilyNameScorePercentage*
- *GivenNameScorePercentage*
- *DateOfBirthScorePercentage*
- *GenderScorePercentage*

- *PostcodeScorePercentage*

Some fields from the response file are carried across in the final data asset created in DPS, depending on the specific data set. In HES, the last 7 fields are carried across as a structure in a field called *MPS Confidence*.

Sensitive flag

Where a patient record is flagged as sensitive, all demographic data is returned in the query response; however, pipeline authors should ensure records are appropriately handled before data is shared with end users, in particular by redacting 'location information' such as addresses which allow individuals to be located.

Error/success code

This code indicates the status of the match at record level. The most common codes will be 00 for success or 98 for not found. See Table 11 for a list of all possible codes.

MATCHED_NHS_NO and *MPS_ID*

These are the fields populated with the matched *NHS number*s if a record was successfully matched in PDS, or with the *MPS_ID* if a record from MPS record bucket was matched instead.

MatchedAlgorithmIndicator field

This field indicates which tracing step was run last before exiting MPS. The complete list of values is listed in Table 3**.**

Note that this field does not indicate whether the record was matched or not. For example, in HES *MatchedAlgorithmIndicator* can assume values 1 or 4 even without a match. This can happen if the record does not satisfy the eligibility criteria to proceed to the next tracing step. The eligibility criteria for cross-check trace are that *NHS number* and *DOB* are present and valid. For algorithmic trace there should be sufficient valid fields to populate at least one block.

Table 3. Explanation of the *MatchedAlgorithmIndicator* field values

Matched Algorithm Indicator value	Explanation
0	None of the tracing steps were run. This can happen when the record does not have sufficient information to run any tracing step, for example when *DOB* is invalid or not available. But there are other edge-case scenarios like unexpected data store errors (for example, the PDS record has duplicates).
1	The last tracing step run was cross-check trace (either DPS or Spine version).
3	The last tracing step run was alphanumeric trace (never run in HES).
4	The last tracing step run was algorithmic trace.

The remaining fields (*MatchedConfidencePercentage, FamilyNameScorePercentage, GivenNameScorePercentage, DateOfBirthScorePercentage, GenderScorePercentage, PostcodeScorePercentage*) are the scores from the different tracing steps.

Scoring in cross-check trace and alphanumeric trace

If a PDS record was matched by cross-check trace, the scoring is binary, that is, *MatchedConfidencePercentage* is either 100 if there was a successful match or 0 otherwise, and *MatchedAlgorithmIndicator* would be 1. This is valid for both DPS and Spine cross-check trace steps. However, the records matched in DPS have null values for the individual percentage scores (for example, *DateOfBirthScorePercentage*, etc), while those matched in Spine have zero values.

Alphanumeric trace is not used for matching HES records, due to the absence of *family name* fields. Its output would also be binary, 100 or 0, depending on whether there was a successful match or not. *MatchedAlgorithmIndicator* would be 2, and all other fields would have zero values.

Scoring in algorithmic trace

In algorithmic trace, the score is a value between 0 and 100, which is the unweighted average of the scores across the name instance (a combination of *given name, other given name and family name* as explained in chapter 3.5), *postcode*, *DOB*, and *gender*, if present. For HES data, *family name* and *given name* are not present so this score will be the average of the scores from the remaining three fields (*postcode*, *DOB*, and *gender*).

FamilyNameScorePercentage and *GivenNameScorePercentage* have values of zero in the MPS outputs but will not contribute to the calculation of the average.

The details of how the scores are calculated can be found in chapter 3.5.

Scoring thresholds

There is no minimum score threshold for a match. However, the blocking rules make sure that there are no candidates with a score below 50. Some users might want to apply further filtering to select only matches with higher confidence score. This can be done in post-processing by using the field *MatchedConfidencePercentage.*

Examples

Table 4 shows examples of scoring outputs for different matching conditions.

If the match was found with algorithmic trace, as in the example in Table 4 column 4, the scores would be as follows; f*amily name* and *given name* were not present (as per all HES examples), but *DOB*, gender, and *postcode* can all match between 0 and 100.

In the MPS_ID matching example (Table 4, column 5), the *MatchedAlgorithmIndicator* can still have values 1 or 4 for HES (other data sets might also have values of 3) depending on which tracing step was last run.

The example in Table 4 column 6 considers when multiple *NHS number*s are matched for the same record. This can only happen with algorithmic trace and therefore the *MatchedAlgorithmIndicator* can only indicate a value of 4 while the remaining scores will be set to 0.

Finally, the example in Table 4 column 7 shows what happens when no match was found (neither in PDS nor in MPS record bucket). The *MatchedAlgorithmIndicator* in this case can indicate any possible value (for HES the only possibilities are 0, 1, 4).

Table 4. Examples of the values that the MPS output fields can assume depending on the tracing step that the record was matched on

Example matched step	DPS cross-check trace	Spine cross-check trace	Alphanumeric trace	Algorithmic trace	MPS_ID matching	Multiple *NHS number* match	No match
MATCHED_NHS_NO	NHS num	NHS num	NHS num	NHS num	0000000000	9999999999	0000000000
MPS_ID	Empty	Empty	Empty	Empty	Present	Empty	Empty
MatchedAlgorithmIndicator	1	1	3	4	1, 3, 4	4	0, 1, 3, 4
MatchedConfidencePercentage	100	100	100	50*-100	0	0	0
FamilyNameScorePercentage	null	0	0	0-100	0	0	0
GivenNameScorePercentage	null	0	0	0-100	0	0	0
DateOfBirthScorePercentage	null	0	0	0-100	0	0	0
GenderScorePercentage	null	0	0	0-100	0	0	0
PostcodeScorePercentage	null	0	0	0-100	0	0	0

* The lower bound of the *MatchedConfidencePercentage* range is not fixed at 50. However, the blocking rules used in the algorithmic trace always guarantee a minimum level of match that practically never produces overall scores below 50.

3.9 Person_ID creation and data set enrichment

The MPS response file contains the fields detailed in Appendix 5.2, including scoring outputs as explained in Chapter 3.8. The response file still does not contain the *Person_ID* field, but only the matched *NHS number* and *MPS_ID*. As part of the DPS pipeline, *Person_ID* is created from these other fields.

The derivation logic for the *Person_ID* is as follows:

- if the record in the response file contains a valid *MATCHED_NHS_NO*, this is used as *Person_ID* (see example in Table 5, row 1)
- else, if the record in the response file contains an *MPS_ID* (*UPRI 1*), this is used as *Person_ID* (see example in Table 5, row 2 - in this case, *MATCHED_NHS_NO* has a default value of 0000000000)
    - if the *MPS_ID* field contains multiple *MPS_ID*s, only the first one is retained (see example in Table 5, row 3)
- else, a *one-time-use ID* (*UPRI 2*) is generated, and this is used as *Person_ID* (see examples in Table 5, rows 4 and 5)

Notably, *MPS_ID* and *one-time-use ID* have a leading letter that helps the users understand the origin of the *Person_ID* (that is, “A/B” if it originates from *MPS_ID*, “U” if it originates from the *one-time-use ID*). These, however, are lost with the tokenization process.

Table 5. Examples of how the fields *MATCHED_NHS_NO* and *MPS_ID* from the response file are combined to produce a *Person_ID*

Row number	*PERSON_ID*	*MATCHED _NHS_NO*	*MPS_ID (UPRI 1)*	*One-time-use ID (UPRI 2)*
1	0123456789	0123456789	-	-
2	A987654321	0000000000	A987654321	-
3	A123454321	0000000000	A123454321~~~ A987656789	-
4	U123123123	9999999999	-	U123123123
5	U312321321	0000000000	-	U312321321

FICTICIOUS DATA

The response file has 34 fields which include personal identifiable information retrieved from PDS. However, not all of them are used to enrich the data set. In HES, only *Person_ID* and the 7 MPS indicators are returned in the data set for the users to see. These are:

- *MatchedAlgorithmIndicator*
- *MatchedConfidencePercentage*
- *FamilyNameScorePercentage*
- *GivenNameScorePercentage*
- *DateOfBirthScorePercentage*
- *GenderScorePercentage*
- *PostcodeScorePercentage*

However, other data sets might have different requirements and return additional personal identifiable information for the patient to enrich the input data set. It is the responsibility of the pipeline author in DPS to add the fields returned by MPS to the original input data.

Superseded *NHS numbers*

An NHS number can be superseded in PDS, which means that it is no longer valid, and it has been replaced by another one. If a query record contains a superseded NHS number, all three tracing steps run in Spine (cross-check trace, alphanumeric trace and algorithmic trace) are capable of recognizing this, and they return the corresponding valid NHS number. This can be confusing for users as they might see a Person_ID that is different from the submitted NHS number being matched at cross-check trace. When this happens, it is most likely a case of superseded NHS number, as illustrated in case study 11 in chapter 4.2.

3.10 Tokenization

In the previous chapters it was established how the *Person_ID* is a unique identifier for each individual patient, generated via the MPS. For HES, this identifier replaces the legacy HES_ID field.

Most users will not have visibility of the clear values for personal identifiable information such as *Person_ID*, but will only have access to the tokenized version of such information, depending under which Data Sharing Agreement (DSA) the data set is provided.

Tokenization (also referred to as “de-id”) is the service that allows for data items to be anonymised. Currently, this happens for *NHS number*, *Person_ID* and *local patient identifier*.

A separate store/table contains the random relationships between the original entry identifier and the token. Tokenized values can also be re-identified using the table in the inverse order.

The token maintains the same format of the original entry so that the same operations can be applied to both fields. In HES, for example, a tokenised version of *Person_ID* can be found in the field *Token_Person_ID*, they are both alphanumeric strings, but *Person_ID* contains 10 digits only, while *Token_Person_ID* contains 32 digits (see example in Table 6).

Notably, the tokenized version of *Person_ID* will no longer preserve the A/B/U initial digit which identifies whether the *Person_ID* was a *UPRI 1* or *2*.

Across the same data set, a unique *Token_Person_ID* is used to identify records with the same underlying *Person_ID*. The same will be true across different data sets, provided that they belong to the same domain. Different domains might be used in different DSAs, and consequently the same *Person_ID* might be identified with different *Token_Person_ID* across different domains.

Table 6. Examples of *NHS number* and *Person_ID* and the tokenized version

NHS number	Person_ID	Token_Person_ID
0123456789	0123456789	987A543219876G432198RR5432198765
-	A123456789	647A5142198RRT432198RHH432112332
-	U123456789	1017JJ146HH8R12322198RHYY771KK32

FICTICIOUS DATA

4. Case studies

To facilitate HES users in their understanding of a real application of MPS, this chapter lists a series of real examples (re-enacted with fictitious data) from HES data sets.

4.1 Aggregated findings from real data

We used the HES data set from the financial year 2021/2022. This is provisional data, final published HES data for 2021/2022 year may differ.

Table 7. Counts of the different combinations of the MPS output fields for HES APC (FY=2021/2022)

MatchedAlgorithmIndicator	MatchedConfidencePercentage	PostcodeScorePercentage	DateOfBirthScorePercentage	GenderScorePercentage	Count
0	0	0	0	0	233,033
1	0	0	0	0	668
1	100	null	null	null	20,666,061
1	100	0	0	0	72,512
4	100	100	100	100	19,516
4	0	0	0	0	25,615

In Table 7 all possible combinations of values for the MPS output fields are displayed for HES APC data set.

With the knowledge of chapter 3 in mind, we are now able to explain the different counts.

There are only 3 unique valid values for the *MatchedAlgorithmIndicator* (that is, 0, 1 and 4) which confirms that no alphanumeric trace step is run for HES, due to the absence of name fields.

The third row shows that most of the matches happen at cross-check trace in DPS, which is expected because HES is a well curated data set where most records will have correct *NHS number*s and *DOB* that match PDS.

72,512 records are still matched with cross-check trace but in Spine instead. This can be seen by the zero (rather than the null) values for the score percentage columns. As we can appreciate in the next paragraph, this can happen for several reasons:

- the match could not be found in PDS cached, but can be found in PDS live
- the *DOB* was only partially correct, and therefore the match was picked up by cross-check trace in Spine which does some additional checks with respect to the step in DPS

- the *NHS number* in the HES record was superseded by another *NHS number*

668 records have a *MatchedAlgorithmIndicator* value of 1 but *MatchedConfidencePercentage* value of 0, meaning that the algorithm could not find a match and it exited at cross-check trace (in Spine) because records did not meet the eligibility criteria for proceeding to algorithmic trace, that is, having valid values in the *DOB*, *gender* and *postcode* fields. These records might either be matched at MPS_ID matching step or not at all.

25,615 records have a *MatchedAlgorithmIndicator* value of 4 but *MatchedConfidencePercentage* value of 0, meaning that MPS attempted all tracing steps without finding a match. These records might either be matched at MPS_ID matching step or not at all.

19,516 records were matched with algorithmic trace, and these have *MatchedAlgorithmIndicator* value of 4 and *MatchedConfidencePercentage* value of 100.

The 233,033 records in row 1 have not been processed at all against PDS because they did not meet the minimum requirements (due to invalid *DOB* field). These might either be matched at MPS_ID matching step or not at all.

4.2 Empirical examples

The examples in this section are based on real results observed in the processing of HES records through MPS, with personal identifiable information all replaced by consistent fictitious values, hence making it impossible to identify real individuals.

Please note that as the HES data set does not include patient names, these examples do not include any instances of alphanumeric trace, or the use of names in algorithmic trace.

These examples use the field names for an MPS request and response files as listed in Table 9 and Table 10, respectively. The equivalent HES field names for readers familiar with HES data are listed in Table 8 below.

Table 8. MPS request file fields

MPS field name in the request file	HES field name they are mapped to
NHS number	NEWNHSNO
Gender	SEX
Date of birth	DOB
Postcode	HOMEADD
Local_Patient_ID	Combination of PROCODET/PROCODE5/PROCODE3 and LOPATID

Group A: happy path scenarios

The first 4 case studies are common happy path scenarios where a *Person_ID* was found.

Case Study 1: valid *NHS number*, matched by DPS cross-check trace

Given a HES record with the following values:

HES record	NHS number	Gender	Date of Birth	Postcode
1	3333333333	2	2000-02-22	LS1 4AP

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	3333333333	1	100	null	null	null

FICTICIOUS DATA

The *MatchedAlgorithmIndicator* value of 1 indicates that a match was found at the cross-check trace step. The null score percentages indicate that it was in the DPS cross-check trace. This corresponds to row 3 in Table 7, where we see is the most popular scenario.

If the score percentages were zero, it would indicate that the match was found at the cross-check trace stage in Spine.

Case Study 2: wrong or null *NHS number*, matched at algorithmic trace

Given a HES record with the following values:

HES record	NHS number	Gender	Date of Birth	Postcode
1	4444444444	2	2000-02-22	LS1 4AP

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	3333333333	4	100	100	100	100

FICTICIOUS DATA

The *MatchedAlgorithmIndicator* value of 4 indicates that a match was found at the algorithmic trace step. If the *NHS number* and *DOB* had been correct, it would have been matched at cross-check trace instead.

The *DateOfBirthScorePercentage* is 100, indicating that it must have been a wrong *NHS number* that prevented it from being matched at cross-check trace. The other score percentages are all 100 which indicates that the returned *Person_ID* is the *NHS number* of a record which matches on *DOB*, *gender*, and *postcode*.

Case Study 3: no *local patient ID*, matched to an existing MPS record

Given a HES record with the following values:

HES record	NHS number	Local patient ID	Gender	Date of Birth	Postcode
1	null	null	2	2000-02-22	LS1 4AP

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	A123456789	4	0	0	0	0

FICTICIOUS DATA

The *MatchedAlgorithmIndicator* value of 4 and *MatchedConfidencePercentage* value of 0 indicate that algorithmic trace was performed but was not successful, so the record was sent on for MPS matching.

The *Person_ID* begins with ‘A’, which indicates that there was a successful match at MPS matching (it could begin with ‘A’ or ‘B’).

Because the input query had no *local patient identifier and* no *given name* or *family name*, it must have matched on *DOB*, *gender*, and *postcode*.

Case Study 4: with *local patient ID*, matched to an existing MPS record

Given a HES record with the following values:

HES record	NHS number	Local patient ID	Gender	Date of Birth	Postcode
1	null	98A21B	2	2000-02-22	LS1 4AP

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	A123456789	4	0	0	0	0

FICTICIOUS DATA

The *MatchedAlgorithmIndicator* value of 4 and *MatchedConfidencePercentage* value of 0 indicate that algorithmic trace was performed but was not successful, so the record was sent on for MPS matching.

The *Person_ID* begins with ‘A’, which indicates that there was a successful match at MPS matching or that a new MPS_ID was created in the MPS record bucket (it could begin with ‘A’ or ‘B’).

Because the input query has a *local patient identifier,* but no *given name* or *family name*, if it was matched it must have matched on local *patient identifier* and *DOB.* We don't know whether it matched on *gender* or *postcode*.

Group B: different identifiers linked to the same *Person_ID*

The next 6 examples have been chosen to show how records with different identifiers can be assigned the same *Person_ID* (and therefore the same *Token_Person_ID*).

Case Study 5: Two records with the same *DOB*, one without *NHS number*, return the same *Person_ID*

Given two HES records with the following values:

HES record	NHS number	Gender	Date of Birth	Postcode
1	3333333333	2	2000-02-22	null
2	null	2	2000-02-22	LS1 4AP

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	3333333333	1	100	null	null	null
2	3333333333	4	100	100	100	100

FICTICIOUS DATA

For HES record 1 we can see a match was made in the cross-check trace step as indicated by the *MatchedAlgorithmIndicator* value of 1. We can infer that an exact match was possible with the *NHS number* and *DOB* provided (as shown by the fact that the same *DOB* scored 100 for HES record 2). Given that a match in the cross-check trace step was made for HES record 1 the scores for *DOB*, *gender* and *postcode* can be either null or zero as they were not calculated in the algorithmic trace step. If null (like in this case), the match was found by cross-check trace in DPS.

In contrast, for HES record 2 which was missing an *NHS number*, the record was matched in the algorithmic trace step, as shown by the *MatchedAlgorithmIndicator* value of 4. In this instance, the patient was matched to the highest scoring PDS record with a score of 100 due to an exact match on *DOB*, *gender* and *postcode*.

As matches to PDS were found for both records, the *Person_ID* field assumes the value of the *NHS number*.

Case Study 6: Two records with different *postcodes* return the same *Person_ID*

Given two HES records with the following values:

HES record	NHS number	Gender	DOB	Postcode
1	3333333333	1	1994-02-24	SW1A 2AA
2	3333333333	1	1994-02-24	SW1A 2AH

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	3333333333	1	100	null	null	null
2	3333333333	1	100	null	null	null

FICTICIOUS DATA

For these records, the fact that the *postcodes* differ makes no difference in the matching process as both contain accurate *DOB* and *NHS number*. This allows the matching to take place at the cross-check trace step.

Case Study 7: Two records with slightly different *DOB*, one without *NHS number,* return the same *Person_ID*

Given two HES records with the following values:

HES record	NHS number	Gender	DOB	Postcode
1	3333333333	1	1982-03-04	SW1A 2AA
2	null	1	1982-03-09	SW1A 2AH

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	3333333333	1	100	0	0	0
2	3333333333	4	100	100	100	100

FICTICIOUS DATA

For HES record 1, the patient was successfully traced by the cross-check trace step.

Record 2 was an exact match on *DOB* (trace score = 100). This means that record 1 must not have matched on full *DOB* and must instead have matched at cross-check trace in Spine on partial *DOB* (1982-03) and outcode (SW1A).

For HES record 2, the patient was successfully traced by the algorithmic trace step with the highest scoring PDS record being 100 on *DOB*, *gender* and *postcode*.

Case Study 8: Two records with slightly different *DOB*, same *NHS number*, return the same *Person_ID*

Given two HES records with the following values:

HES record	NHS number	Gender	DOB	Postcode
1	3333333333	1	1976-10-05	LS1 4AP
2	3333333333	1	1971-10-05	ZZ99 3WZ

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	3333333333	1	100	null	null	null
2	3333333333	1	100	0	0	0

FICTICIOUS DATA

The *MatchedAlgorithmIndicator* value of 1 and *MatchedConfidencePercentage* value of 100 indicate that a match was found at the cross-check trace step. The null score percentages for the first record indicate that it was in the DPS cross-check trace. The second record has scores of zero, hence the match was found at the cross-check trace stage in Spine.

In record 2, DPS cross-check trace could not find a successful match because the year of birth is incorrect, however, Spine cross-check allows for a partial *DOB* match where the outcode (the left part of the *postcode*) matches. In this example, the patient had on its postcode history a ‘ZZ99’ which matched the outcode.

Case Study 9: Two records with different *gender* return the same *Person_ID*

Given two HES records with the following values:

HES record	NHS number	Gender	DOB	Postcode
1	3333333333	1	1994-02-24	LS1 4AP
2	3333333333	9	1994-02-24	LS1 4AP

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	3333333333	1	100	null	null	null
2	3333333333	1	100	null	null	null

FICTICIOUS DATA

For these records, the fact that the gender differs makes no difference in the matching process to the person ID as both contain accurate *DOB* and *NHS number*. This allows the matching to take place at the cross-check trace step in DPS.

Case Study 10: Two records with different *gender* and *postcode* return the same *Person_ID*

Given two HES records with the following values:

HES record	NHS number	Gender	DOB	Postcode
1	3333333333	9	1970-07-01	ZZ99 3WZ
2	3333333333	2	1970-07-01	LS1 4AP

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	3333333333	1	100	null	null	null
2	3333333333	1	100	null	null	null

FICTICIOUS DATA

The *MatchedAlgorithmIndicator* value of 1 and *MatchedConfidencePercentage* value of 100 indicate that a match was found at the cross-check trace step. The null score percentages for the first record indicate that it was in the DPS cross-check trace. *Postcode* and *gender* are different in the two records; however, this does not affect the cross-check trace behaviour because it only looks at *NHS number* and *date of birth*.

Group C: edge cases

The final 4 examples have been chosen to demonstrate edge cases, where it is helpful to explain some results where a match could not be found, or which may appear surprising.

Case Study 11: Two records with superseded versus current *NHS numbers*

Given two HES records with the following values:

HES record	NHS number	Gender	DOB	Postcode
1	4444444444	2	2003-03-03	null
2	5555555555	2	2003-03-03	LS1 4AP

FICTICIOUS DATA

We would receive the following response fields following processing in MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	4444444444	1	100	null	null	null
2	4444444444	1	100	0	0	0

FICTICIOUS DATA

For HES record 1, we can see a match is made in the cross-check trace step within DPS as indicated by the *MatchedAlgorithmIndicator* value of 1. This means that an exact match was possible using the *NHS number* and *DOB* provided by the PDS records cached within DPS.

HES record 2 has a different *NHS number* but is matched to the same *Person_ID* as HES record 1 and has the same *MatchedAlgorithmIndicator* value of 1. *DOB*, *gender* and *postcode* score percentage fields have values of 0, which means that record 2 was cross-check traced in Spine. The success of the tracing to a *Person_ID* associated with a different *NHS number* allows us to infer that 555555555 is an invalid *NHS number* which has been superseded by 4444444444. Cross-check trace in DPS does not return matches where *NHS number*s are superseded, while cross-check trace in Spine was able to recognize the match because it also checks for superseded *NHS number*s.

Case Study 12: Two records with different *gender*, *postcode, and NHS number* return the same *Person_ID*

Given two HES records with the following values:

HES record	NHS number	Gender	DOB	Postcode
1	4444444444	1	1994-02-24	SW1A 2AA
2	3333333333	2	1994-02-24	SW1A 2AH

FICTICIOUS DATA

We could receive the following response fields following processing through MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	3333333333	1	100	0	0	0
2	3333333333	1	100	null	null	null

FICTICIOUS DATA

This is the same as case study 11, except that the *gender* and *postcode* are different in the two records, superficially leading a HES user to believe that these would be two different patients sharing the same date of birth. However, MPS assigns both to the same *NHS number* via cross-check trace. The reason is that cross-check trace does not use *gender* or *postcode*, and if *NHS number* 444444444 was superseded by 333333333, then Spine cross-check trace would be able to pick these up as the same *Person_ID*.

Case Study 13: *One-time-use ID* generated as no matches at any stage

Given a HES record with the following values:

HES record	NHS number	Gender	Date of Birth	Postcode
1	null	1	2003-03-03	LS1 4AP

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	U123KE3ABC	4	0	0	0	0

FICTICIOUS DATA

The *MatchedAlgorithmIndicator* value of 4 with *MatchedConfidencePercentage* of 0 indicates that none of the trace steps returned a match against the PDS records or the MPS record bucket. This could indicate that there are no individuals which match the identifying characteristics provided, or that multiple matches were returned and MPS was unable to determine a single match.

This record has valid *DOB* and *postcode*, so it contains sufficient information to create a new *MPS_ID*. However, the *Person_ID* begins with 'U' indicating that a *one-time-use ID* was generated for this record, hence we conclude that multiple NHS numbers were matched, and algorithmic trace could not resolve the match.

Case Study 14: Invalid *DOB* results in no matches at any stage

Given a HES record with the following values:

HES record	NHS number	Gender	Date of Birth	Postcode
1	3333333333	2	1800-01-01	LS1 4AP

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	U123KE3ABC	0	0	0	0	0

FICTICIOUS DATA

The *MatchedAlgorithmIndicator* value of 0 indicates that all steps were skipped. This is because every trace step requires *DOB*, but in this case the *DOB* was recognised as invalid.

The *Person_ID* begins with 'U' indicating that a *one-time-use ID* was generated for this record, hence there were no sufficient information to even generate a new *MPS_ID*.

Case Study 15: Same Person_ID for records corresponding to different people

Given three HES records with the following values:

HES record	NHS number	Local patient ID	Gender	DOB	Postcode
1	3333333333	D012347	1	1880-01-01	ZZ99 3WZ
2	4444444444	F123458	1	1880-01-01	ZZ99 3WZ
3	5555555555	H234569	1	1880-01-01	ZZ99 3WZ

FICTICIOUS DATA

The following response fields could be returned by MPS:

HES record	Person_ID	Matched Algorithm Indicator	Matched Confidence Percentage	Date Of Birth Score Percentage	Gender Score Percentage	Postcode Score Percentage
1	B123456789	4	0	0	0	0
2	B123456789	4	0	0	0	0
3	B123456789	4	0	0	0	0

FICTICIOUS DATA

The *MatchedAlgorithmIndicator* value of 4 with a *MatchedConfidencePercentages* of 0 indicates that the records ran through 2 cross-check trace steps and algorithmic trace. However, it did not find a match as this stage. The 3 records were matched to the same *MPS_ID* during the MPS_ID matching phase.

It is unlikely that these 3 records, with different *NHS numbers* and *local patient IDs* refer to the same person, but because of the use of pseudo postcodes in conjunction with default values for *DOB*, MPS_ID matching is not able to distinguish between them.

Notably, the *DOB* for this example is not recognized as invalid by MPS. If it were, then the *MatchedAlgorithmIndicator* would have been 0, and the three records would have each been assigned a *one-time-use ID.*

4.1 Mps_diagnostics can help with the case studies

The examples laid out in the case studies may be confusing to analysts when encountered in real data, since the *Person_IDs* are accompanied by little (or no) information on the matching process. Some relevant contextual information can be found in the MPS response record (returned by MPS), however this record is not usually shared with users, including on HES, where only the *MatchedAlgorithmIndicator* and the confidence scores are available. The mps_diagnostics data set was created to address this shortcoming.

MPS Diagnostics is the pipeline that produces mps_diagnostics and uses the contextual information from the MPS response file, and some additional data from PDS, to create 10 columns of metadata explaining in user-friendly terms how each *Person_ID* was derived.

An accompanying document explains more about mps_diagnostics, revisiting some of the case studies in this chapter and demonstrating how mps_diagnostics helps to explain them.

mps_diagnostics is available upon request for internal NHS England analysts via CDAs (clear data agreements), or for external NHS E users via DSAs (data sharing agreements).

5. Appendix

5.1 Request file

Table 9. Request file field descriptions

Column name	Description
UNIQUE REFERENCE	Identifier of the request record
NHS_NO	NHS number
FAMILY_NAME	Surname
GIVEN_NAME	First name
OTHER_GIVEN_NAME	Other names
GENDER	Gender (sex) of the person
DATE_OF_BIRTH	Date of birth. The date format is comprised by: YYYY – 4-digit year MM – 2-digit month DD – 2-digit day
DATE_OF_DEATH	Date of death
ADDRESS_LINE1	First line of the person's address
ADDRESS_LINE2	Second line of the person's address
ADDRESS_LINE3	Third line of the person's address
ADDRESS_LINE4	Fourth line of the person's address
ADDRESS_LINE5	Fifth line of the person's address
ADDRESS_DATE	The date from which the address has been indicated to be valid
POSTCODE	Postcode of the person's address
GP_PRACTICE_CODE	GP practice code
NHAIS_POSTING_ID	Unique code that represents the NHAIS box
AS_AT_DATE	The date the data should be valid for
LOCAL_PATIENT_ID	Local patient ID used by providers
INTERNAL_ID	Internally used ID. This field is used for the value that is provided in the data set and will be different depending what type of data has been submitted in the request file. For the HES data set it will contain the HES_ID field
TELEPHONE_NUMBER	Person's telephone number
MOBILE_NUMBER	Person's mobile number
EMAIL_ADDRESS	Person's email address

5.2 Response file

Table 10. Response file field descriptions

Column name	Description
UNIQUE REFERENCE	Identifier of the request record
REQ_NHS_NO	Requested NHS number provided in the request file data record, if any
FAMILY_NAME	Surname
GIVEN_NAME	First name
OTHER_GIVEN_NAME	Other names
GENDER	Gender (sex) of the person
DATE_OF_BIRTH	Date of birth
DATE_OF_DEATH	Date of death
ADDRESS_LINE1	First line of the person's address
ADDRESS_LINE2	Second line of the person's address
ADDRESS_LINE3	Third line of the person's address
ADDRESS_LINE4	Fourth line of the person's address
ADDRESS_LINE5	Fifth line of the person's address
ADDRESS_DATE	The date from which the address has been indicated to be valid
POSTCODE	Postcode of the person's address
GP_PRACTICE_CODE	GP practice code
NHAIS_POSTING_ID	Unique code that represents the NHAIS box
AS_AT_DATE	The date the data should be valid for
LOCAL_PATIENT_ID	Local patient ID used by providers
INTERNAL_ID	Internally used ID. This field is used for the value that is provided in the data set and will be different depending what type of data has been submitted in the request file. For the HES data set it will contain the HES_ID field.
TELEPHONE_NUMBER	Person's telephone number
MOBILE_NUMBER	Person's mobile number
EMAIL_ADDRESS	Person's email address
SENSITIVE FLAG	S = Sensitive / Y = Legacy Sensitive / I = Invalid / N = Legacy Not Sensitive / B = Legacy Under Business Investigation
MPS_ID	Also known as *UPRI 1* (which stands for unmatched person record identifier) Unique *MPS_ID* created by MPS
ERROR/SUCCESS_CODE	See Table 11
MATCHED_NHS_NO	'1234567890' = valid matched NHS number '0000000000' = no matched NHS number

	'9999999999' = multiple NHS numbers matched OR not enough data (e.g., if DOB or postcode is not provided)
MatchedAlgorithmIndicator	0 = None of the tracing steps were run 1 = Cross-check trace was the last step run 3 = Alphanumeric trace was the last step run 4 = Algorithmic trace was the last step run
MatchedConfidencePercentage	Total % score (0 or 100% for cross-check and alphanumeric, weighted and aggregated average for algorithmic)
FamilyNameScorePercentage	Score derived from the family name comparison between query record and PDS records. Not available in HES. See chapter 3.5 for more details.
GivenNameScorePercentage	Score derived from the given name comparison between query record and PDS records. Not available in HES. See chapter 3.5 for more details.
DateOfBirthScorePercentage	Score derived from the DOB comparison between query record and PDS records. See chapter 3.5 for more details.
GenderScorePercentage	Score derived from the gender comparison between query record and PDS records. See chapter 3.5 for more details.
PostcodeScorePercentage	Score derived from the postcode comparison between query record and PDS records. See chapter 3.5 for more details.

5.3 Error and success codes

Table 11. Codes for ERROR/SUCCESS_CODE field in the response file and the possible *Person_ID* types it can be associated with.

Note only one error code will be returned, even if a record falls due to more than one criterion.

Error Code	Person_ID
00 (Success)	NHS number/ MPS_ID/ One-time-use ID
05 (Unexpected error)	One-time-use ID
11 (Field length exceeded)	One-time-use ID
12 (Invalid gender values)	One-time-use ID
13 (Invalid field format)	One-time-use ID
14 (Data store error)	One-time-use ID
15 (No trace performed)	One-time-use ID
16 (Not enough fields provided)	One-time-use ID
17 (Number of fields greater than allowed)	One-time-use ID
90 (Success superseded)	NHS number
91 (Invalid)	One-time-use ID
92 (Sensitive)	NHS number/ One-time-use ID
96 (Not Enough Data)	One-time-use ID
97 (Multiple Matches)	One-time-use ID
98 (Not found)	MPS_ID/ One-time-use ID

5.4 Soundex Matching

Soundex is a method of representing the sounds of names as strings. The output of Soundex is used to compare Given and Family names in alphanumeric and algorithmic trace. The output isa letter followed by three numbers.

The Soundex of a name is calculated as follows: first, any non-ASCII characters are removed. Then, each letter in the name is mapped to a number as described in Table 12. The Soundex output consists of a letter followed by 3 numerical digits: the letter is the first letter in the name. The subsequent digits are Soundex mapped numbers of the name, with consecutive digits removed and cropped to the first 3 digits. In the case where the Soundex number are fewer than 3 digits then zeros are padded from the right-hand side. The process is summarised in Figure 9.

Examples of Soundex computation

Example 1: Consider the name Mary. The scoring is calculated as the following:

- the name, Mary is converted to Soundex numbers as 5060
- the zero digits are removed and it becomes 56
- the first number of the left-hand side is replaced with the first letter in Mary thus becomes M6
- the Soundex number has fewer than 4 characters thus the final output is padded with zeros on the right-hand side to become M600

Example 2: Consider the name Mary-Janet. The scoring is calculated as the following:

- the hyphen, a non-ASCII character, is removed for the name to become MaryJanet
- the name, MaryJane is converted to Soundex numbers as 506020503
- the zero digits are removed and it becomes 56253.
- the first number of the left-hand side is replaced with the first letter in MaryJane thus becomes M6253
- the Soundex number has more than 4 characters thus the final output is cropped the first 4 characters form the left-hand side to become M625

Example 3: Consider the name Fábián. The scoring is calculated as the following:

- the letters with acute accent, a non-ASCII character, are removed for the name to become Fbin

- Fbin is converted to Soundex numbers as 1105
- the duplicate consecutive numbers are removed thus 105
- the zero digits are removed and it becomes 15
- the first number of the left-hand side is replaced with the first letter in Fábián thus becomes F5
- the Soundex number has fewer than 4 characters thus the final output is padded with zeros on the right-hand side to become F500

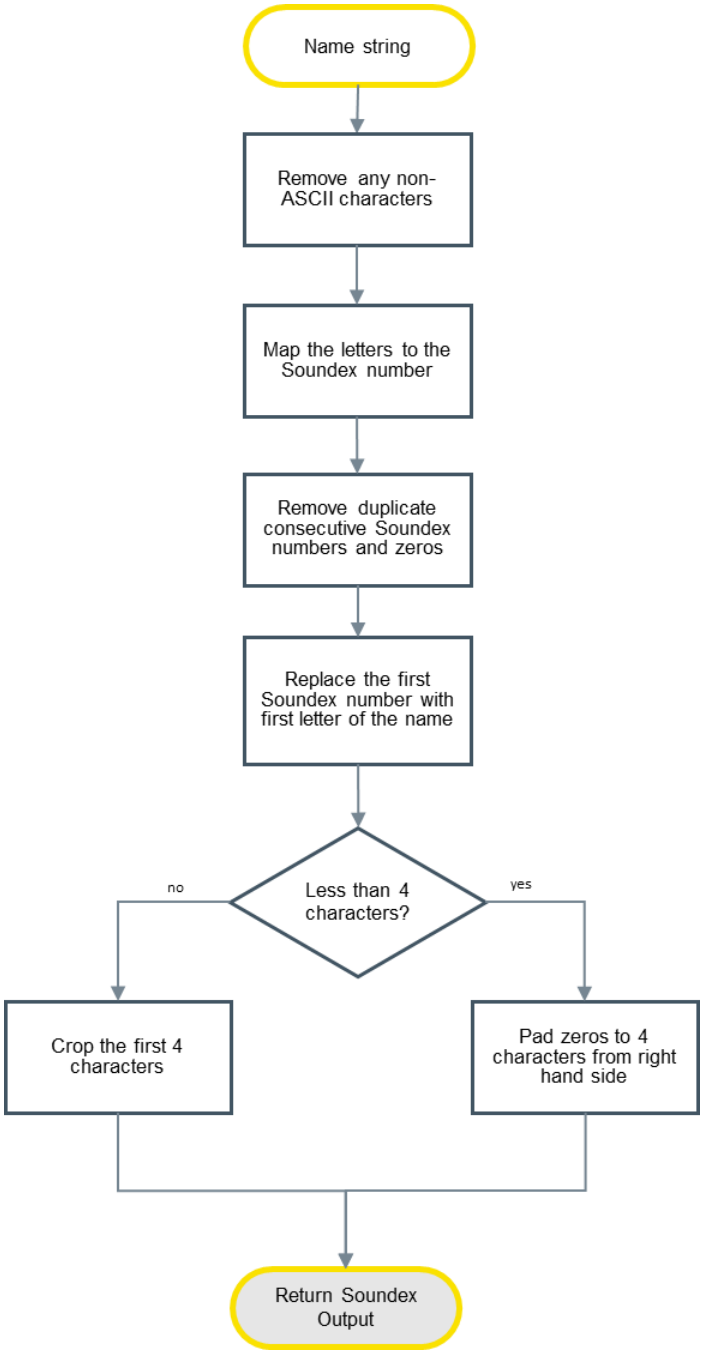

Figure 9: Flowchart of computing the Soundex score

Table 12: Soundex Coding for each letter

Letter	Soundex Coding
A	0
B	1
C	2
D	3
E	0
F	1
G	2
H	0
I	0
J	2
K	2
L	4
M	5
N	5
O	0
P	1
Q	2
R	6
S	2
T	3
U	0
V	1
W	0
X	2
Y	0
Z	2