# Change

Updated 16th December 2018 to 1.3.2Main Changes

1. Correct getLocalPid to return the matched LPID where one is found

# EMPI Data Structures

The EMPI schema is a separate schema from the repository.

## Person

A representation of the key demographics of the incoming record, uniquely identified by local ids - type/org/number (“MRN” / “RXX01” / “12345”).

This record will contain standardized fields to support algorithmic tracing and can hold historic data although currently only the previous surname is supported for performance reasons.

## MasterRecord

A national master record. There will be a Master Record for every NHS Number, CHI Number, H&SC, UKRR and UKRDC Number. This is uniquely identified by type and id (e.g. “NHS” / “1234567890”)

## LinkRecord

A simple x-reference table between the Person and MasterRecord. This identifies verified links between a person and a master record.

Uniquely identified by personid and masterid.

To find all records for a UKRDC record you would need a query such as:

select \* from jtrace.person p, jtrace.linkrecord lr, jtrace.masterrecord mr

where lr.masterid = mr.id

and lr.personid = p.id

and mr.nationalid = 'RR3000001'

and mr.nationalidtype = ' UKRDC '

## Relationship between Masters

The relationship in this model between two master records potentially referring to the same physical person is always indirect. There is no direct relationship between an NHS Number and a CHI Number except through a Person record claiming to link the two numbers – essentially because that link is the view of the source system not a definitive link recognized by NHS England and the NHS Scottish.

At first glance this might seem an impediment but the creation of UKRDC Master record and links to the associated source system Person records removes that complexity for real life access. Once created the UKRDC Master becomes the “top of the tree” accessing related person records as required.

## Links to the Repository

Records in the Repository are in a different schema but keyed by the same local ids as used in the person table in the EMPI. This provides for an easy link between the two.

## PIDXRef

The PIDXref table manages the consolidation of local patient records which arise due to local circumstances which cause a single patient from a center to present with more than one patient id. Examples being Cardiff where records can present with or without a suffix to the MRN or centers which consolidate data from more than one issuing authority.

The PIDXref acts as a pre-deduplication process which means that the core EMPI only needs to manage 1 single validated local patient identity for each center and can focus on matching local records to national records.

PIDXref croff-references the generated PID – which is effectively a consolidated local patient id or CLPID – with the key data coming in from the center (Sending Facility, Sending Extract, and Local Id).

The PID on the PIDXref is then the id that is used throughout the EMPI (and the UKRDC Repository)

# Client

The Client (the repository) will create a Person record containing

* Local Ids
* Primary Id (if present - “UKRDC” + UKRDC Number)
* National Ids (a list of other national ids present on the RDA)
* Demographics

It will call EMPI Store passing in the Person

# API: Validate

## Returns

UKRDCIndexManagerResponse

## Behaviour

Call ValidateInternal

Call ValidateAgainstEMPI

On exception create and return a FAIL response including the error message and stack trace.

Otherwise return a SUCCESS response including the national identity.

## ValidateInternal

* If entered the primary id must be UKRDC
* The surname must be at least 2 characters
* The given name must be at least 1 character
* Gender must be at least 1 character
* Date of Birth must be provided
* Local Id must be provided – at least 1 character
* Local Id Type must be provided
* Originator must be provided

## ValidateAgainstEMPI

If skip duplicate check option is set then return. (skip is set to allow “fake” MRNs sent by RADAR)

For each national Id on the inbound record that already exists in the EMPI

Count links to this Master from this Unit (Originator) - excluding current record

* + If any exist – reject the record

# API: Store

## Returns

UKRDCIndexManagerResponse

## Behaviour

Call Create or Update

On exception create and return a FAIL response including the error message and stack trace.

Otherwise return a SUCCESS response including the national identity.

## Create Or Update

If effective date is not provided then default to today.

**Call ValidateInternal**

**Call Standardise**

Find the person for the inbound record (originator, local id type, local id)

If found

**Update**

Else

**Create**

## Standardise

Left and right trim and convert to upper case

* Given Name
* Other Given Name
* Surname
* Title
* Gender
* Postcode
* Street

Convert postcode to standard form

* Remove any embedded spaces
* If postcode length < 5 – do nothing (invalid)
* If postcode length >7 – do nothing (invalid)
* Insert a space before last 3 characters

## Create

NormalizeSurname

NormalizeGivenName

NormalizePostcode

Insert Person record

\*\* Maintain the links to other national ids

For each National Id on the inbound record (NHS/CHI/H&SI)

**Create National Id Links**

\*\* Maintain the primary index (UKRDC number)

**Create UKRDC Link**

## Update

NormalizeSurname

NormalizeGivenName

NormalizePostcode

If surname has changed

NormalizeSurname passing in the previous surname

Update Person record

\*\* Remove any national identifies which are no longer on the person record

For each LINK on the database for this person

Get the Corresponding MASTER

If not UKRDC

If this national Id & type is also on the inbound record

**Update National Id Links**

Mark this national id as *processed*

If this national Id is not on the inbound record

Delete from the database

If no other links to this MASTER

Delete the MASTER

\*\* Add any new national links

For each National Id on the inbound record (Not marked as *processed*)

**Create National Id Links**

\*\* Maintain the primary index (UKRDC number)

**Update UKRDC Link**

## Normalization

### Normalize Surname

Trim and convert to upper case

Lookup in the Surname homonym list, default to the original name

Trim the result

Calculate the Soundex for the result

Return the soundex

### Normalize Given Name

Trim and convert to upper case

Lookup in the Given homonym list, default to the original name

Trim the result

Calculate the Soundex for the result

Return the soundex

### Normalize Postcode

Trim and convert to upper case

Remove any embedded spaces

## Create UKRDC Link

If a primary id is on the inbound record

Find the MASTER record for that primary id

If found

If **VerifyMatch**

Create LINK

If the effective date on the inbound is > effective date on the MASTER

Update the demographics on the MASTER

If not

Create WORK

Create LINK

update MASTER as INVESTIGATE

If the effective date on the inbound is > effective date on the MASTER

Update the demographics on the MASTER

Else

Create a MASTER record

Create a LINK to it

Else

\*\* Try to find a matching UKRDC record which can be corroborated by other national ids

For each national id on this record (NHS/CHI/H&SI)

Get the MASTER Record by national id

Get all the Links to this MASTER

For each linked record (excluding the inbound person)

Search for a MASTER UKRDC number LINKed to this person

If found

Does the inbound record **VerifyMatch** against this UKRDC Master?

If so

Create LINK

Audit Link

Else

Create WORK

If not linked

Allocate UKRDC Number

Audit Allocation

Create MASTER

Create LINK

## Create National Id Links

Find the MASTER record for the national id details provided

If found

Create LINK

If not **VerifyMatch**

Create WORK

mark MASTER as INVESTIGATE

If the effective date on the inbound is > effective date on the MASTER

Update the demographics on the MASTER

If skip Duplicate Check not set in the API

Check Duplicates for the MasterId/Originator

If found

Create WORK

mark MASTER as INVESTIGATE

Else

Create a MASTER record

Create a LINK to it

## Update UKRDC Links

If a UKRDC number is linked to the inbound record

Find the MASTER record for that primary id

If there is no primary on the inbound OR it is the same as that stored

If the effective date on the inbound is > effective date on the MASTER

Update the MASTER demographics

**Verify Links**

Else

// Stale update

If the record does not **VerifyMatch**

Create WORK

Mark MASTER as INVESTIGATE

Else

// Primary has changed

Delete the original LINK

If no LINKs remain for the MASTER

Delete the MASTER

**Create UKRDC Link**

Else

**Create UKRDC Link**

## Update National Id Links

If demographics have changed

If the effective date on the inbound is > effective date on the MASTER

Update the demographics on the MASTER

**Verify Links**

Else

If no longer **VerifyMatch**

Create WORK

Mark MASTER as INVESTIGATE

## Verify Links

// Called when the demographics on the master are updated

For each Person linked to this master (except the person causing this change)

If no longer **VerifyMatch**

Create WORK

Mark MASTER as INVESTIGATE

## VerifyMatch

If DOB Matches exactly

Return TRUE

If 2 parts of the DOB Match

If the first 3 characters of Surname and 1 character of Given name matches

Return TRUE

RETURN FALSE

## GetLocalPid

1. Read PIDXREF by SF/SE/mrn
2. If exists
   1. *This record has already been linked to this PID*
   2. **Return the PID** (as PIDXREF.PID)
3. If doesn’t exist
   1. For each National Id (NI) on the inbound record
      1. Look for NI linked to a local id for this SF AND SE (requires PIDXREF in the lookup) – In other words looking for situations where only the number has changed.
      2. If found
         1. Verify full demographics (GENDER, DOB, SURNAME, FORENAME from PERSON, not from the EMPI which may include data from other sites for this National ID)
         2. If matched
            1. *This record has not been seen by the EMPI before, but it will link to another local record*
            2. Return the LPID for the matched recod found
         3. If not matched
            1. *This record has not been seen by the EMPI before, but it is related by a NI to another local record with different demographics*
            2. *Create a Work Item with the inbound details, National Id, and reason for mismatch*
            3. **Return “REJECT”** (The caller should reject this record)
   2. If no match found
      1. *This record has not been seen by the EMPI before and no local link is found so it will be allocated a new PID on update.*
      2. **Return “NEW”** (meaning the update call will allocate a CLPID)

## SetLocalPid

1. Read PIDXREF by SF/SE/mrn
2. If exists
   1. [Person will be updated as normal when the store process.]
   2. **Return the PID**
3. If doesn’t exist
   1. For each National Id (NI) on the inbound record
      1. Look for NI linked to a local id for this SF AND SE (requires PIDXREF in the lookup) – In other words looking for situations where only the number has changed.
      2. If found
         1. Verify full demographics (GENDER, DOB, SURNAME, FORENAME from PERSON, not from the EMPI which may include data from other sites for this National ID)
         2. If matched
            1. Insert PIDXREF
            2. Audit the match

SF/SE/mrn linked to existing LocalPID xxxxx

PersonId is the existing person as the new one has not yet been inserted in the person table

* + - * 1. **Return the matched PID**
      1. If not matched
         1. Audit Not Required
         2. Create work item – including the degree of match

Note: PersonId will be 999999999 on the work item because the person record has not been created.

Log the sf/se/mrn and mismatching demographics on the WorkItem as properties

* + - * 1. **Return “REJECT”** – The caller should reject the record
  1. If no match found
     1. Generate next new PID (Sequence Number)
     2. Save PIDXREF with PID & SF / SE / mrn / GENDER / DOB / SURNAME / FORENAME
     3. Audit Not Required
     4. **Return the newly allocated PID**

# API: Search

## Returns

UKRDCIndexManagerResponse

## Behaviour

Call SearchInternal

On exception create and return a FAIL response including the error message and stack trace.

Otherwise return a SUCCESS response including the national identity.

## SearchInternal

// Used to find a UKRDC number from another national id and demographics

Programme Search Request must be provided and must contain a National Id

Find the MASTER for the National Id provided

If not found - return null

For each person LINK to the MASTER

Search for a LINKed UKRDC MASTER

If found

**VerifyMatch** the demographics provided against the UKRDC MASTER found

If verified – return the MASTER

# API: Link

## Returns

UKRDCIndexManagerResponse

## Behaviour

Call LinkInternal

On exception create and return a FAIL response including the error message and stack trace.

Otherwise return a SUCCESS response including the national identity.

## LinkInternal

// API call will be used for a Manual Link

Find a LINK for the master id and person id provided

If exists

Reject

Create LINK

# API: getUKRDCId

## Accepts

Int : MasterId

## Returns

UKRDCIndexManagerResponse

## Behaviour

Calls MasterRecordDAO to retrieve the MasterRecord for the provided MasterID

If not found create and return a FAIL response (“Master ID does not exist”)

If the master is not a UKRDC master, create and return a FAIL response (“Master ID is not a UKRDC ID”)

On exception create and return a FAIL response including the error message and stack trace.

Otherwise return a SUCCESS response including the national identity.

# API: merge

## Accepts

Int : superceedingId

Int : supercededId

## Returns

UKRDCIndexManagerResponse

## Behaviour

Calls LinkRecordDAO to retrieve all links to the supercededId

For each

Delete the link

Create a link to the superceedingId

Audit the event (Type=3, UKRDC\_MERGE – contains superceeding and superceded id’s)

Delete all Work Items for the supercededId

Delete the Master Record for the supercededId

On exception create and return a FAIL response including the error message and stack trace.

Otherwise return a SUCCESS response

# SimpleConnectionManager: Configure

## Accepts

* String : user
* String : password
* String : server
* String : port
* String : dbName
* Int : poolSize (optional, defaults to 10)

## Returns

void

## Behaviour

Sets up the database before the EMPI can be used.

This call establishes the Connection Pool for communicating with the JTrace database. If not provided the poolSize will default to 10.

# WorkItemManager

The WorkItemManager exposes APIs for management of the Work Items. Currently only update is tested and documented but other APIs for creating, finding and deleting items exist and can be added to the documented public interface as required.

## Update

Taken from the javadocs

* + - **update**
    - public [WorkItem](file:///C:\Users\Nick\git\jtrace\JTrace\doc\com\agiloak\mpi\workitem\WorkItem.html) update(int workItemId,
    - int status,
    - java.lang.String updateDesc,
    - java.lang.String updatedBy)

throws com.agiloak.mpi.MpiException

Update the Work Item using the id as the key. Certain values are not updateable as they are intrinsic to the WorkItem (personId, masterId, type). Last updated date will automatically be updated

**Parameters:**

workItemId - - REQUIRED - The id of the WorkItem being updated. This must exist

status - - REQUIRED - The new status of the WorkItem [WorkItemStatus](file:///C:\Users\Nick\git\jtrace\JTrace\doc\com\agiloak\mpi\workitem\WorkItemStatus.html)

updateDesc - - REQUIRED - Description of the work item update

updatedBy - - REQUIRED - Who is updating the item

**Returns:**

The WorkItem following the update

**Throws:**

com.agiloak.mpi.MpiException - For any exception encountered.

### Behaviour

Validates the parameters.

Finds the WorkItem for the id provided – raising an exception if not found.

Updates the retrieved WorkItem with the data provided for:

* Status
* UpdatedBy
* UpdateDesc

And sets the lastUpdated time to current time

Updates the WorkItem in the database

Create and Audit record saving attributes for:

* Id
* Status
* UpdatedBy
* UpdateDesc