The PMI Consolidaton software suite assists with merges, acquisitions and takeovers (of a smaller hospital by a larger hospital) or cleaning existing data before enabling an HL7 interface. However, it can also be used for simple data quality checking (does the data in one system have the right names/dob/sex when compared to another system).

The PMI Consolidation software comes with some documentation; however it might not be all that useful as it does assume a knowledge of patient master indexes and Python.

PMI Consolidation – How to do it is a somewhat academic treatise on what one might aim to achieve in a mergers/acquisitions/takeovers scenario. Or what one must strive to achieve if you are enabling an HL7 feed to a previously standalone application.

Guide to Checking the Files Created by the PMI Consolidation Software is all about what to do when you find PMI errors and how to go about fixing them. You can’t fix everything all at once. You will probably fix some records, and deem other unfixable, and then get new extracts and run the process all over again. There are configuration files that you can use to ensure that you are not being told, every time, about things you already know. If you are not involved in cleaning, then this document is irrelevant.

The PMI Consolidation Software is the most detailed of the three documents. It is almost a user manual. But it assumes a good understanding of PMIs and secondary applications holding their own PMI.

The PMI Consolidation software suite is based upon a slightly complex view of a Patient Master Index. It assumes that every ‘name/dob/sex’ record has its own record identifier – the patient ID or PID. The PID is an internal database concept. If there is no PID in a PMI extract, then you can use the row number in the file as a surrogate for PID values. If PID is the internal identifier, then MRN/UR is the external identifier. The Master Record Number (MRN) or Unit Record number (UR) is the number used to file bits of paper in a compactus of patient records. The terms MRN and UR are interchangeable. The PMI Consolidation software refers to this identifier as UR.

Now every PID record has a UR, but the same UR can occur on more than one PID record. However, only one PID record for each UR will be marked as the ‘primary’ name/dob/sex record. All other PID records for this UR will be marked as aliases. Alias can be created because patients have nicknames. Or they can be created when patients are merged.

The underlying assumption here is that all data is stored against PID values and that retrieving a patient’s data involves assembling the set of PID values for the UR and then retrieving all data stored against any of those PID values. Moreover, it is assumed that all merge history is kept at the PID level to facilitate unmerges.

The PMI Consolidation software suite assumes a similar PMI structure for departmental or service provider applications. However, here the UR number is only public internally to the service provider. It is the number used by the service provider to store their paperwork about each patient. The service provider may have to communicate to multiple hospitals about the same patient, and each time they will have to use the hospital’s UR. So, each PID record in the service provider’s application is assumed to be associated with a list of alternate UR numbers; one AltUR for each hospital.

When a PMI extract is created from a department or service provider’s application, it should/could contain the application specific PID, the service provider’s UR and it must contain the AltUR (hospital’s UR) that the service provider uses when communicating with this hospital. However, this is rarely the case. Sometimes the department is a wholly owned subsidiary of the hospital and uses the hospital UR as the departmental UR. Often, all you get in the department or service provider’s PMI extract is one UR; the hospital’s UR. If this is the case, then you can use record number as a surrogate for PID and UR as a surrogate for AltUR.

The matching process is quite simple. It compares two files; master.csv and secondary.csv. These are cleaned up, simpler versions of the extract. They have a consistent layout, making comparison easier. The first step is to create a dictionary from the master.csv file where the key is the UR number and the value is the concatenation of the familyName~givenName~dob~sex. Then, for each record in the secondary.csv file, check that the AltUR is a key in this dictionary. If the key is found, then retrieve the value and compare that to a concatenation of familyName~givenName~dob~sex from this record in the secondary.csv file. A perfect match is deemed a perfect match.

If you are looking to strip bad data out of, say, a data migration extract, then these perfect matches are all you need. However, if to maximize the number of matches, then you will have to review the imperfect matches; records that match on some, but not all, of familyName, givenName, dob and/or sex. The matching process reports all these imperfect matches, and there is a process for checking them, and identifying imperfect matches that are “close enough”.

The remaining records are deemed non-matches, either because they have no AltUR, or because the AltUR is known to be bad, or because all the imperfect matches were inconclusive. A variant of the matching process is then used to find these patients in the master.csv file; a variant that ignores any AltUR/UR matching and tries to match every non-matched secondary.csv patient with every master.csv patient, based upon familyName~givenName~soundOfFamilyName~soundOfGivenName~dob~sex. It is also possible to include in this finding process things like matching addresses, birthdates that are close, or have the day and month swapped. However, at the end of this process, all you will get is reports of good-looking approximate matches which will need be checked to see which ones are “close enough”. Again, there is a checking process that lets you determine which ones were found and which ones remain non-found.

At the end of this process you will know which secondary.csv records to leave alone, which ones need their AltUR updating and which ones need there AltUR removed, because it is incorrect and there is no correct AltUR for this patient.