PROCOMP
SOFTWARE CONSULTANTS

A different kind of MIS software product for behavioral health organizations.

555 Cinti.-Batavia Pike, Cincinnati, OH 45244 • Phone: (513) 685-5245 • Fax: (513) 685-5293
Toll Free: (800) 783-1668 • Email: catt@procompsoftware.com • www.procompsoftware.com

White Paper – HL7 Bridge Technology

Overview

Catt's Bridge Module offers real-time data sharing between Catt and any other data system. This offers organizations a great deal of flexibility integrating Catt into the organization's IT infrastructure. The HL7 Bridge Module described here offers the capability for a single point of data entry for the basic client information in one of the systems, with an automatic real-time updating or bridging of the data to the other system.

There are three typical data sharing scenarios. The first two involve the use of the Catt clinical desktop as the primary data front end to another backend AR/MIS system. In the first scenario presented below, the data systems run on different IT platforms while the second presents the same functionality with both systems running within a MS SQLServer environment. The third scenario reverses the data flow with Catt serving as the backend AR/Billing and MIS system and another front end data input system or clinical desktop.

Scenario One: Catt Clinical Desktop Front End to Backend Mainframe AR System in Cross Platform Hospital Environment

When the AR/MIS system runs on a radically different infrastructure than Catt's Microsoft Network based SQL Server environment, this real-time data exchange is handled with the use of all three of the HL7 Bridge Module's data processing services. An example of this more complex implementation would be a large hospital environment in which the behavioral health care clinic uses Catt as the clinician desktop with all billing is processed by the hospital mainframe AR system. In this particular example, the basic client residence and demographic information as well as the family insurance and billing information is entered once into the registration screens of the hospital billing system and then automatically appears within a matter of seconds within the Catt system. This is accomplished by three separate simultaneously running software processes involving a "data transporter", "data translator", and "data updator". For each data set or transaction bridged between systems, each component completes it's respective task and passes the data onto the next component in a sequential manner.

TIPS Interface Service: The actual mechanics of this data exchange and the functionality of each of the three software services making up the HL7 Bridge Module is presented below. During this discussion we will be referring to screen graphics presented in the Appendix to this

paper. In the hospital based example, the data exchange is accomplished via an industry standard data file specification called "HL7". Most mainframe hospital software systems can be set up to automatically "broadcast" or publish a data record immediately following each "new record" or "update" operation using anyone of hundreds of specific HL7 "transactions" or data sets (e.g. patient address or demographic information entered in the registration screen, insurance company and policy information whenever a new payor is added or updated, movement information whenever a new clinic or doctor assignment is made). The transport or "broadcasting" of these transactions can be handled in a variety of ways but is performed most commonly by industry standard communication software packages such as "OpenLink". These communication systems establish "point-to-point" connections between two computer systems and merely transfer data between them, much like a modem transfers data packets over a phone line or the internet. This first communications or data transport step within Catt is handled by the **"**TIPS Interface Service**"**. This software component is a standalone VB.Net application that is installed on a dedicated Bridge workstation as an operating system "service". This establishes the "point-to-point" physical connection with the hospital OpenLink server at a specific IP address and Port Number (configured via registry keys on the dedicated Bridge workstation). The service runs continuously (as does all three bridge components) listening for any available transactions being broadcast by the OpenLink server. Any incoming transactions (e.g. records of a particular data set) are written to Catt's "Raw HL7 Interface" table. A screen graphic of the viewer utility for this table is presented on Page 9 of the Appendix.

HL7 Parser Service: The second Catt Bridge Module component used in this complex cross platform, hospital-based data bridging example is called the **"**HL7 Parser Service**"**. Like the first component, this is a standalone VB.Net application that is installed on the same dedicated Bridge workstation as an operating system "service". This software program continuously polls the Raw HL7 Interface Table looking for transaction records newly posted by the TIPS Interface Service (e.g. the first component). When it finds one, the transaction string is parsed into single data elements accordingly to the specifications of that particular HL7 transaction. While the HL7 data specification is supposed to be generic and standard for all users and all transaction types, in reality this is not the case. These specifications in Catt are thus user definable via the "HL7 Parser Configuration" table. A screen graphic of the user interface for this table is presented on Page 8 in the Appendix. As each data item or field in each particular transaction is extracted, it is written to a Catt HL7 Interface table. A separate interface table is deployed for each type of HL7 transaction imported from the broadcasting system. In one children's hospital example, four different types of transactions are imported and posted to four transaction tables: patient demographics, parent/relative demographics, insurance payors for each patient account, and a list of primary care physicians.

HL7/Catt Update Service: The third Bridge component is the **"**HL7/Catt Update Service**".** This software procedure is built into the main Catt executable and was designed to also run continuously, usually on the same dedicated bridge workstation as the other two OS based services. The update service continuously loops through each HL7 Interface table looking for newly written records by the parser service (e.g. component two). When translated data records are found, each field in the record is written to the live Catt production data tables. This update

service is driven by a user configurable table called the "HL7/Catt Interface Map". A screen graphic of the user interface for this table is presented on Page 9 of the Appendix. For each data item to be bridged, the operator creates a record in this configuration table, mapping a data column from the "source" table (one of the HL7 Interface tables), to a data column in the "destination" table (a live Catt production data table). Extensive date stamping of all updates are maintained in the "LastEdit" fields of the live production tables. In addition, each time the update service updates a set of fields in a live production table, an "archive" record is added to the source HL7 Interface table containing the previous values of all updated data fields. This allows updates to be rolled back on a case-by-case basis.

Workflow Processing Issues: With all three HL7 Bridge Module services running continuously, anytime a user adds or updates any of the data elements in the hospital system which have been configured for bridging, a HL7 transaction is broadcast through the OpenLink communication system and the sequence of events described above is triggered. As each service completes its task, the next service is activated, with the end result being that the Catt production tables are updated with the broadcasted data elements. The speed at which this transfer occurs depends primarily upon three factors: the delay between completing a data addition or change in the hospital mainframe and the time it takes for OpenLink to broadcast the transactions, the speed of the communications pipe between the OpenLink server and Catt's SQL Server, and finally the "Update Service Interval" setting on the HL7 Interface administration screen (screen graphic presented on Page 6 of the Appendix). This last setting determines how often the update service polls the HL7 Interface tables for newly received and translated records. With an Update Service Interval setting of 5 seconds in an environment in which direct point-to-point IP communications occurs across a local LAN connection, newly edited information can appear within Catt in as little as 5-10 seconds after the operation is completed in the mainframe system.

The mechanics described above allow for the single entry of general client information to drive both the AR/Billing operation in one data system and the clinician front end in another. When using multiple data systems in which several functional capabilities exist in both, users must decide which functions (e.g. registration, scheduling, rostering, QA and editing checking) are best deployed with each system. Such decisions depend upon a host of factors, mostly functional pros and cons in each area between system designs, agency workflow and efficiency issues. For these different operations to run effectively, certain system and MIS data elements must also be coordinated between the systems (such as staff codes, service codes, locations, and etc). While these tables can also be bridged allowing for single entry, this can be at times very difficult given the differing data structures that drive a system's picklists or constants as well as the data translation that may have to occur during the synchronization process. For simplicity sake, it is most common that all automated and synchronized data through the Bridge Module flows in a single direction. In the hospital example, the data that is synchronized is always edited in the hospital system with these data elements in Catt always updated automatically. There is of course a great wealth of information that is entered and maintained directly in Catt, that the mainframe AR system has no need for.

The other major exchange of data occurs when a clinician logs an episode of service from within the Catt clinical desktop, and that episode log is translated and exported for AR processing by

the backend AR/Billing application. While this can also be sent out real-time by the Bridge Module as each episode log is finalized by the clinician, more often the receiving AR/Billing system cannot import real-time transactions. For these systems, episode logs created by clinicians are exported as a batch in a file specification which the receiving AR/Billing system can then import.

ARBatch Processor: This batch exporting functionality is not part of the Bridge Module but a service offered by the ARBatch Processor which is an integral part of Catt's billing system. This utility coordinates all of Catt's batch transaction or record processing facilities including monthly statements and insurance forms, creation of ANSI837 EDI transaction files or MACSIS Behavioral Health date, data exports to OLAP data analysis systems and the like. On a routine schedule, back office data clerks will run the episode log export routine that will create a text file containing all episode logs recently finalized by clinicians. This routine is programmed to perform any data transformations (e.g. data type conversions, mis code translations, etc) required for importing into the backend AR system for posting to the receivables system. Also as part of this routine, extensive data validity and QA checks can be performed to insure the accuracy of the exported data. Depending upon the agency's workflow and service/encounter volume, these batch data exchange processes can be run either daily, multiple times per day, or weekly.

Scenario Two: Catt Clinical Desktop Front End to Backend AR System in a common SQL Server Environment

The second scenario described here involves the very same functionality as scenario one but requires a much simpler implementation of the Bridge Module. This is allowed by fact that both data systems (Catt's clinical desktop front end and the other backend AR/Billing system) run in the same networked MS SQLServer IT environment. If the data for each of these two systems reside on a MS SQLServer, only one of the three Bridge Module services is necessary to achieve the real-time data interfacing. If the two data systems either reside on the same server or on different servers within the same security domain, the individual data tables from both systems can be "linked" to the same Catt executable providing real-time or direct access to the data in both systems. With this architecture, there is no need for the data transporter service. Because the client demographic and residence information or AR payor data is usually available in separate fields or columns in the standard table format of SQL Server, there is no need for the second Bridge Module service, the data translator. In this scenario, the only Bridge Module service that is necessary is the third, the data updator. As discussed earlier, the HL7/Catt Update Service is driven by a configuration table called "HL7/Catt Interface Map". For each data item to be bridged, the operator creates a record in this configuration table, mapping a data column from the "source" table to a data column in the "destination" table. While in scenario one, the source tables were the HL7 Interface tables populated by the data translator process, in this scenario, the source tables are the actual live production tables in the backend AR data system.

There are two necessary conditions required for this more simplified bridge architecture to work. Each of the tables in the backend AR data system containing data fields to be bridged must contain some sort of date stamp column which is updated each time the data in the table has been

changed by the backend data system. This provides the update service with a way of coordinating the data exchange. The second requirement is that no data transformations are necessary in moving the data between the two systems. If this is the case, the second Bridge Module service (the data translator), will needed to provide this.

Scenario Three: Third party clinical front end interfacing with Catt's backend AR/MIS Module.

This scenario is the reverse of the first two scenarios presented above. In this case, episode encounter information is created by some other clinical front end or automated data collection schema (such as PDA's or paper scanning technologies) and then imported into Catt for processing and posting to Catt's AR system. This usually occurs as a batch process that is driven by the ARBatch Processor without the need for the Bridge Module services. Batches of encounter records are imported on some sort of scheduled basis and then processed by the Episode Log Posting utility, in an identical fashion as episode logs created from within Catt's clinical desktop are processed to billing. QA and edit checking can be preformed on this imported data at this time.

The three Bridge Module services would be deployed in a similar fashion as in the first scenario if real-time data exchange is implemented for patient demographics, residence or other general account information. This data can be transferred in either direction.

Support Services Required for Implementing Bridge Services

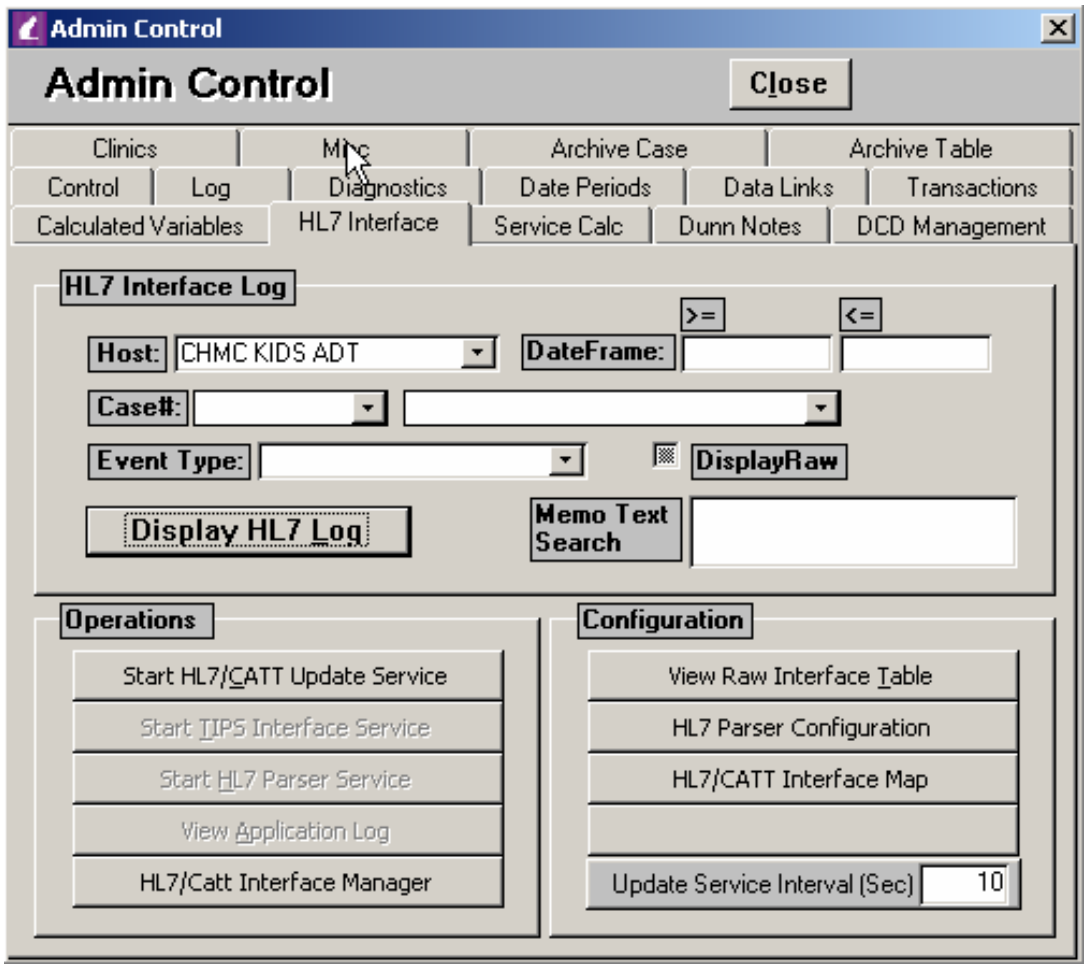
The number of support hours required for implementing this bridge technology depends primarily upon the type and infrastructure of the external data system, ProComp's prior experience with bridging data from that infrastructure and of course the number of data fields, number of separate data tables and the direction of the data exchanges.

As suggested by the three scenarios presented above, when real-time data exchange via use of the Bridge Module is not implemented, and the only data that is exchanged is encounter information via the ARBatch Processor, only minimal support services will be required to program a single batch process to either export or import the episode log table. If real-time data exchange is required between Catt and another SQL Server based data system, the support services will be limited to the time required to configure the HL7/Catt Interface Map and test the HL7/Catt Update Service.

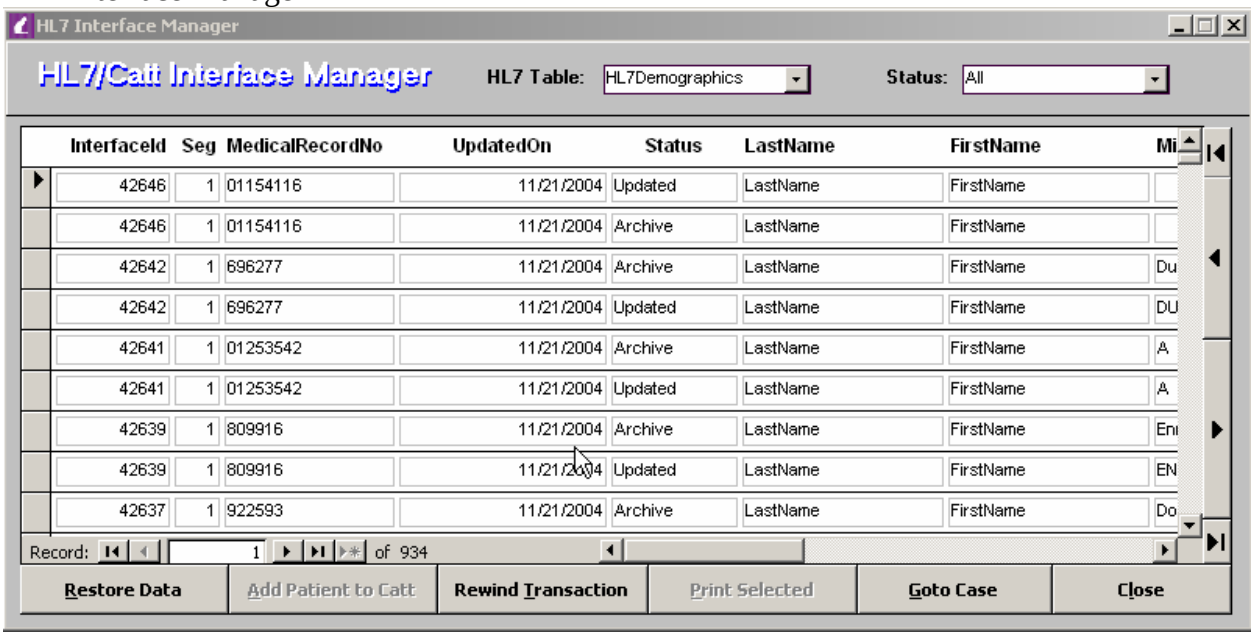
Implementing the Bridge Module services to bridge information from a different data infrastructure often requires extensive support services. Each of the first two Bridge Module services may have to be customized or developed from scratch depending upon the both the transport mechanism and the file structure. Even the HL7 Parser server may require some coding changes depending upon the particular HL7 transactions that need to be translated.

Appendix

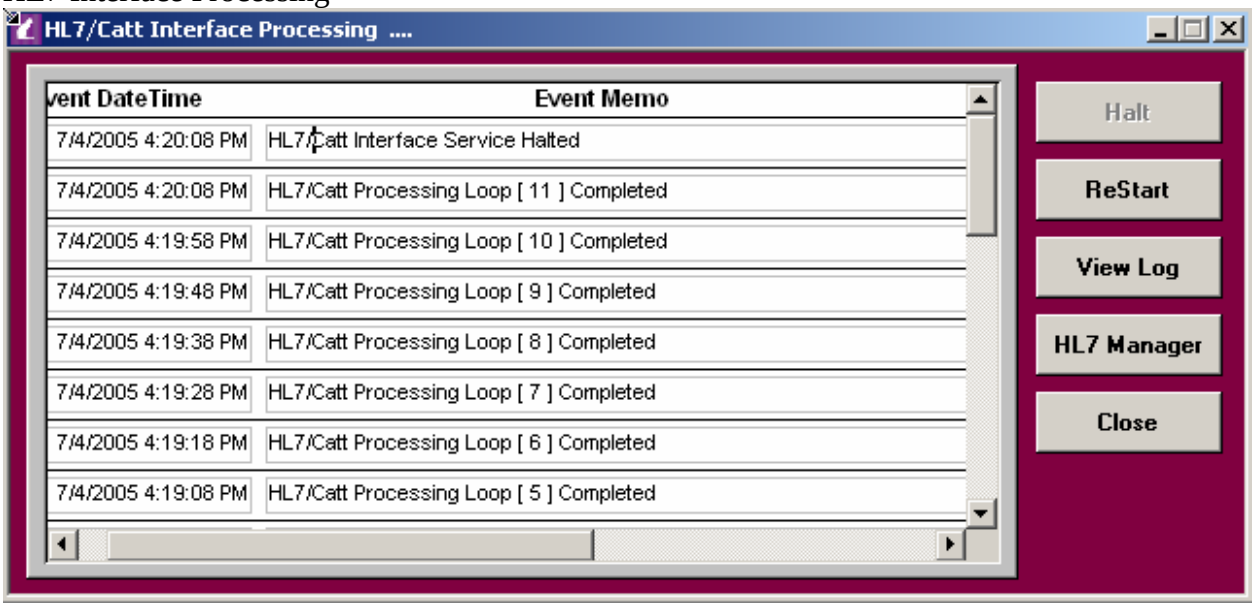
HL7 Bridge Module Administration Console



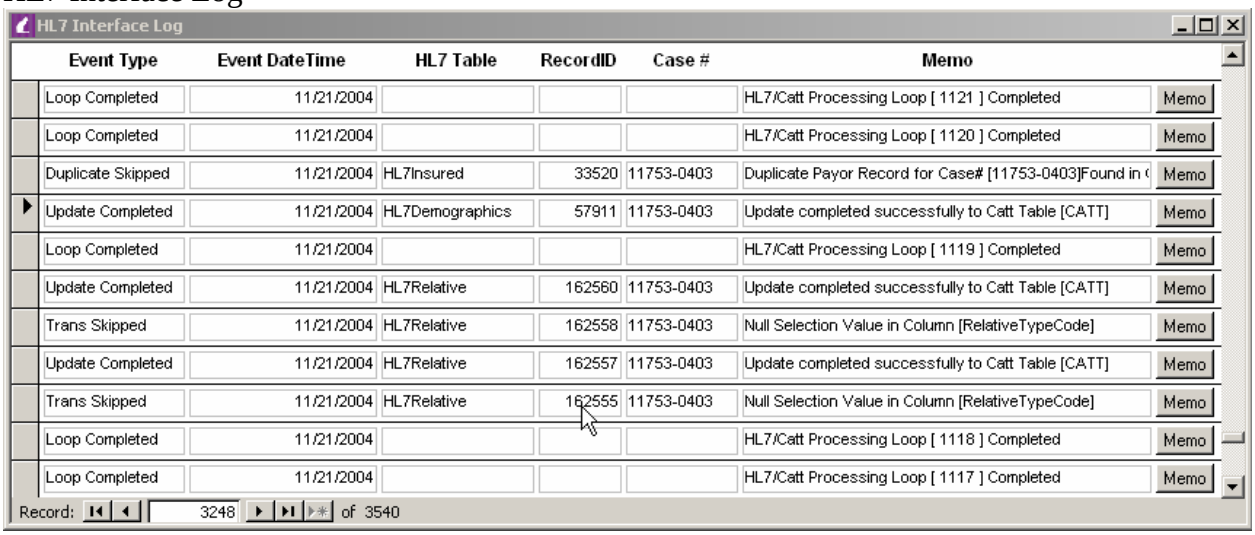
HL7 Interface Manager



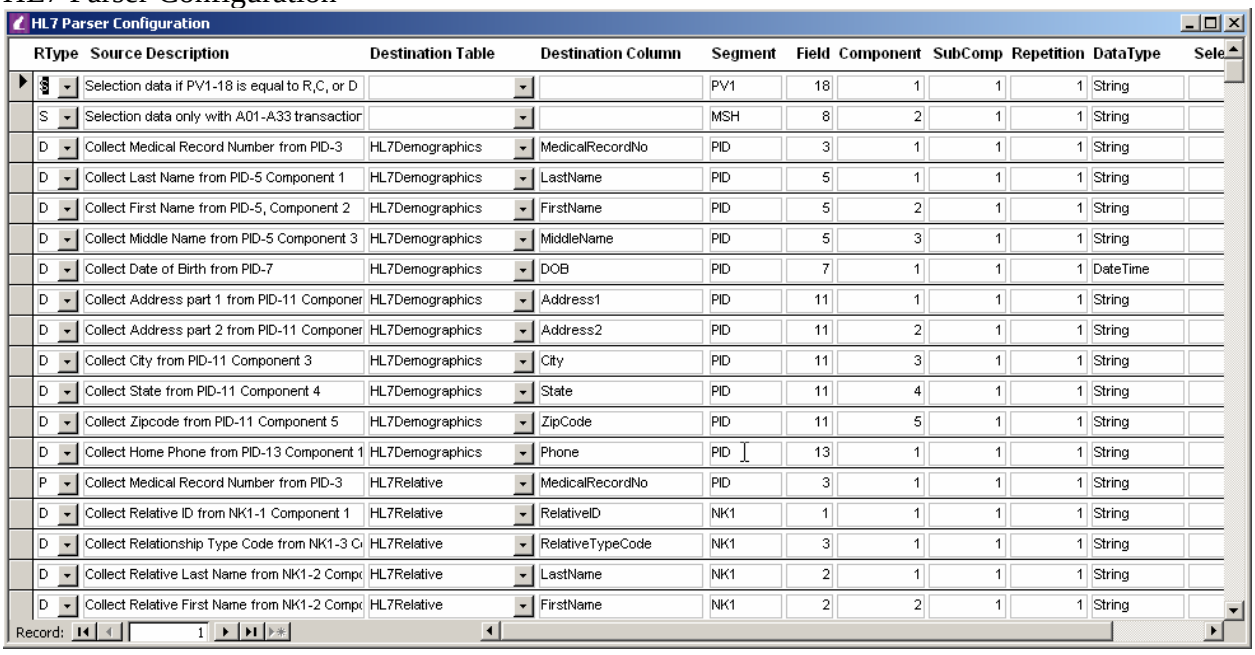
HL7 Interface Processing



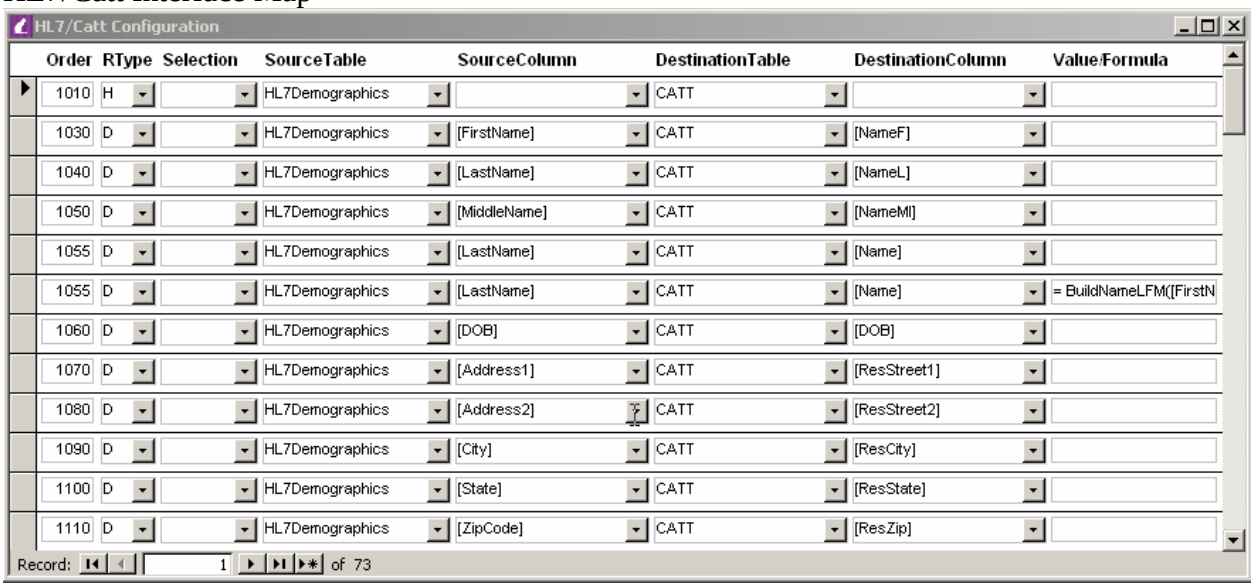
HL7 Interface Log



HL7 Parser Configuration



HL7/Catt Interface Map



Raw HL7 Transactions