VERMONT
AGENCY OF HUMAN SERVICES

briljent

State of Vermont
VHCURES and VHIE Integration Strategy

Prepared for:

State of Vermont

Kristin McClure

Mahesh ThopaSridharan

Bechir Bensaid

Prepared by:

Briljent, LLC

Laura J. Larimer – llarimer@briljent.com

Ariel Jimenez

Heidi Fox

Susan Clark

Due: September 29, 2022

This page is intentionally left blank.

Table of Contents

Introduction **2**
Current State **2**
VHCURES 2
VHIE 3
Future State **5**
Data Modeling, Design, and Standardization 5
Data Pipeline and Ingestion Processes 6
Data Migration 7
Risks and Challenges **7**
SaaS to SaaS 8
Patient Identifiers 8
Patient Matching 8
Parallel Procurements and Projects 9
Roadmap **9**
Next Steps **9**
Appendix **11**

List of Figures

Figure 1: VHCURES System Workflow 3
Figure 2: VHIE System Workflow 4
Figure 3: Proposed Integrated Data Flow 5

List of Tables

Table 1: Roadmap of activities by objective 9
Table 2: VHCURES data dictionary with hashed fields highlighted 11

Introduction

The State of Vermont desires an integrated data set combining the data of their All-Payer Claims Database (APCD), the Vermont Health Care Uniform Reporting and Evaluation System (VHCURES), and the Vermont Health Information Exchange (VHIE), which provides clinical data for care coordination and analytics. Furthermore, the vision is to store this combined data in VHIE which will become the focal point of their analytics layer in the Vermont Unified Health Data Space (VUHDS).

The purpose of this document is to provide an integration plan which will outline the current state of the two systems, their future desired state, and the steps needed to achieve the future state. Any known risks will be identified, and mitigation strategies will be proposed. The result will be an actionable plan for the State to execute to implement the desired integration.

As several other procurements and projects are active in parallel with this one, some of the steps may need to be adjusted or tailored to work in conjunction with them. However, the concepts introduced in this document should be general and flexible enough to coordinate with any parallel projects and still be implemented in a timely manner.

This document only focuses on the technical aspect of integrating the data. Please refer to the Health Data Strategy and Data Governance Framework for information on those topics.

Current State

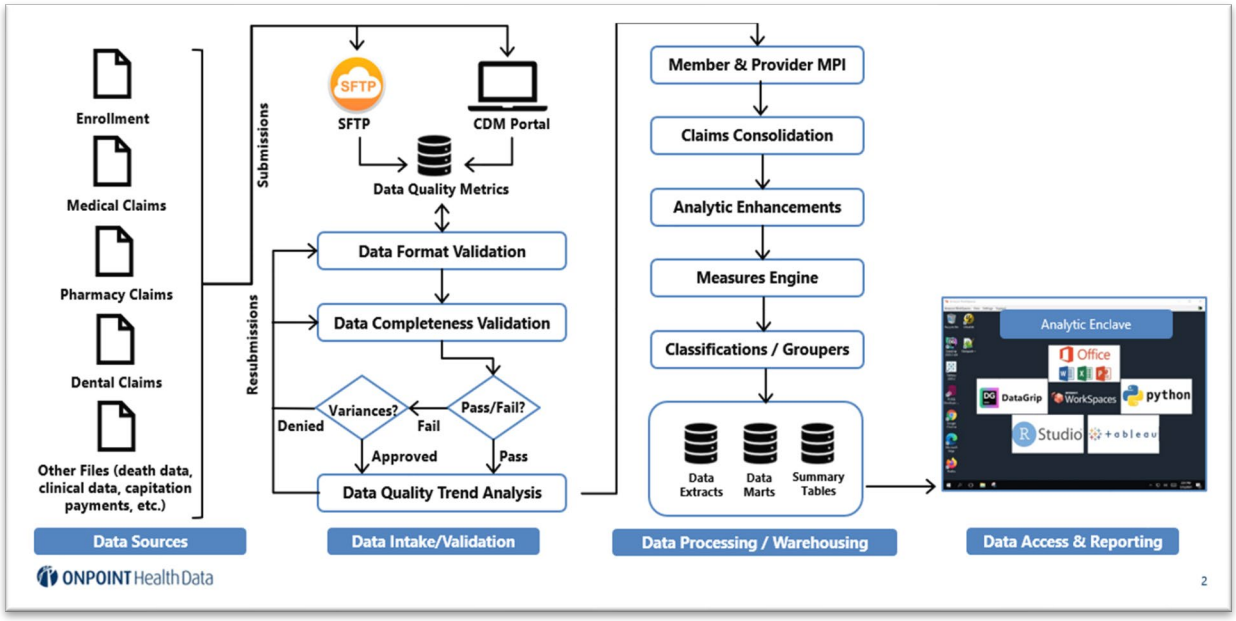
The following describes and illustrates the existing data flow occurring in VHCURES and VHIE.

VHCURES

VHCURES ingests claims and enrollment data from Medicare, Medicaid, and commercial payers. This data provides information such as plan and member payments, Current Procedural Terminology (CPT) codes, International Classification of Diseases (ICD) diagnoses codes, date of service, and servicing provider. The data set does not include clinical data elements such as referral information, test results, medications, immunizations, allergies, vital signs, encounters, smoking status. The data only includes information on paid claims for insured individuals, and does not include uninsured persons or persons who paid out-of-pocket

VHCURES is run on a Software-as-a-Service (SaaS) platform managed by Onpoint Health Data. VHCURES is hosted by Amazon Web Services (AWS) as a Redshift data warehouse.

Figure 1: VHCURES System Workflow



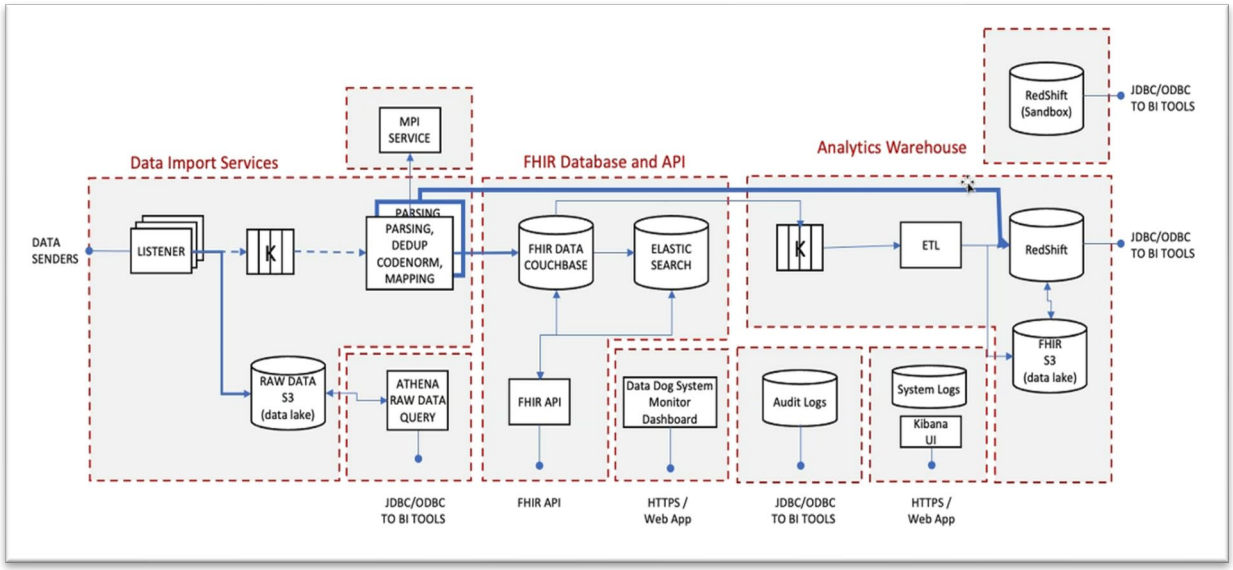
The data ingestion process for VHCURES follows a standard data flow. Claims files are submitted by the payers using Secure File Transfer Protocol (SFTP) or the Web Portal. These files must be submitted with hashed Protected Health Information (PHI) data such that the individual claimants are not recognizable in the data file. A validation process occurs in VHCURES once the files are submitted to validate proper formatting. Any file that does not meet the formatting requirements is immediately rejected for correction by the submitter. Accepted files are passed through an extract, transform and load (ETL) process which involves checking against a Master Person Index (MPI) to assign an existing ID or create a new ID if the person is not yet in the MPI. The data is cleansed and stored in the data warehouse where it is made available for analysis in the Analytic Enclave, a virtual Windows environment providing analytic tools and access to the stored data for approved users. Any query results, reports, or extracts that need to be saved from the Analytic Enclave can be placed in an AWS S3 bucket accessible to the data user.

Since the PHI data of the claims file is required to be hashed prior to submission, the VHCURES data does not have demographic data from which to perform patient matching with external datasets.

VHIE

The VHIE data warehouse currently contains clinical data from various data sources. This data includes lab result, pharmacy, physician, provider, home health, etc. This information is provided through HL7 messages and integrated through Rhapsody. VHIE is run on the Medicasoft NXT platform which is hosted in AWS and managed by Vermont Information Technology Leaders (VITL). The VHIE data warehouse utilizes Redshift.

Figure 2: VHIE System Workflow

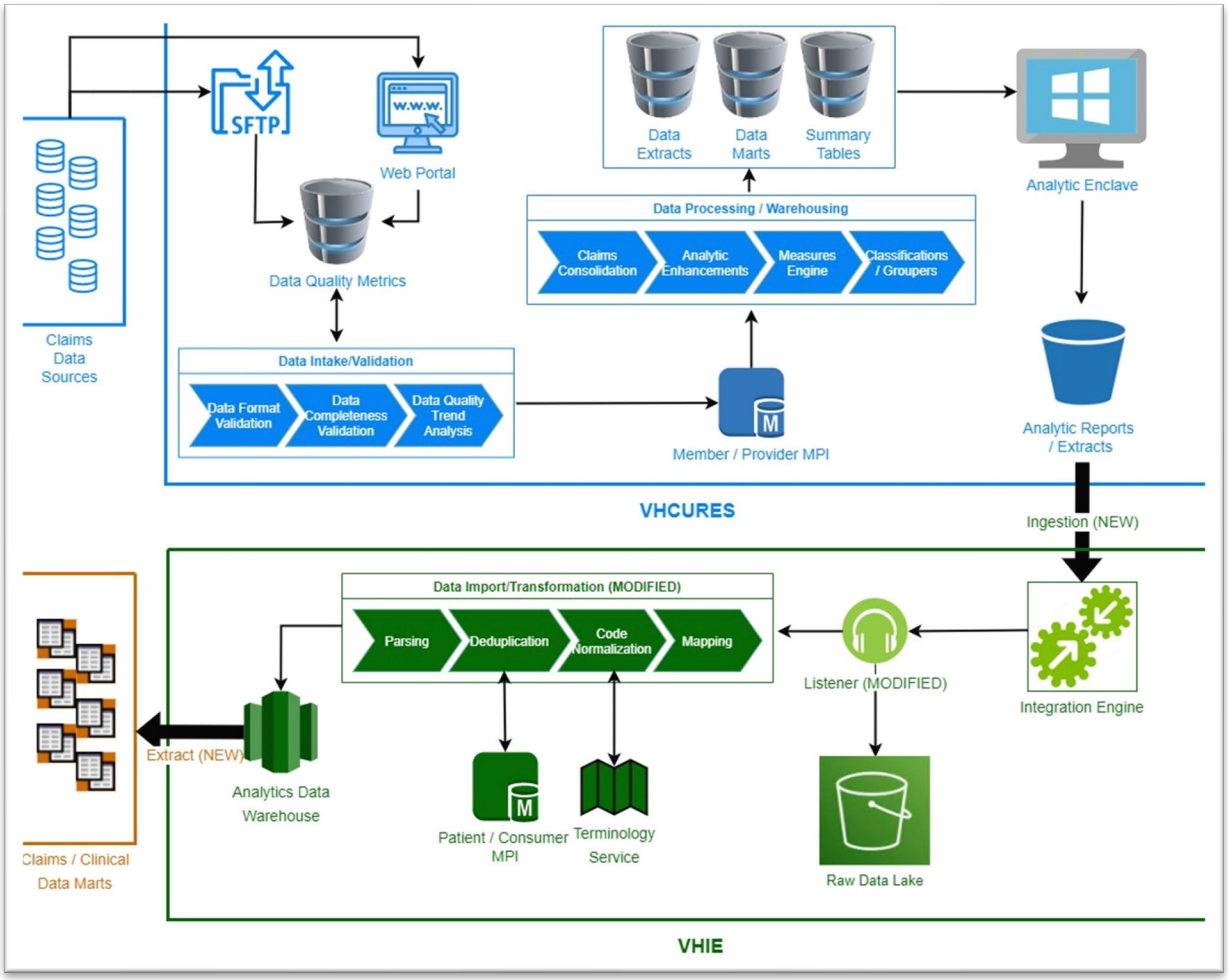


The VHIE data ingestion process begins with Health Level 7 (HL7) messages integrated through the Rhapsody engine. A listener process retrieves the message and sends it through a message queue to an ETL process while simultaneously saving to a data lake. The ETL process cleans the data through a process of parsing, deduplication, code normalization, and mapping. The ingested data is first parsed to determine what type of data (test results, encounter information, discharge notes, etc.) to process it correctly. As part of this process, the data enters a deduplication process to check for duplicate messages since the live HL7 feeds may send the same message several times. It also checks for duplicate patient records against the Verato MPI to prevent multiple records for the same patient. The data then enters a code normalization phase in which the information is translated through a terminology service to standardize the coding for data storage. The data is then stored as a Fast Healthcare Interoperability Resources (FHIR) resource in a document database and made accessible via FHIR Application Programming Interface (API). The data is then sent through another message queue and ETL process to break down the FHIR resource into assembly tables for storage in the Redshift analytics data warehouse. At this point, the data is available for analysis through standard Business Intelligence (BI) tools using a Java Database Connectivity/Open Database Connectivity (JDBC/ODBC) connection.

Future State

The State of Vermont desires an integrated data set of claims and clinical data stored within the VHIE analytics data warehouse. This data set will be made available to the Data Analytics Reporting (DAR) solution which is currently in procurement. The following key components must be developed to achieve this state.

Figure 3: Proposed Integrated Data Flow



Data Modeling, Design, and Standardization

The following goals of data modeling, design, and standardization are discussed below.

- Identify data sets to be exported
- Establish claims database in analytics warehouse
- Perform data modeling for schema design of claims database
- Standardize claims data with respect to existing clinical data
- Execute successful proof-of-concept (POC) for integration of claims data

Identify data sets to be exported

The applicable data sets from the VHCURES database will need to be identified for export based on the State's anticipated use cases. If data fields from the new source database are duplicated in the target database, it may help to tag each record with an indicator of the source (such as adding another field) to help resolve the duplicates. Please note that this is required if the exported data will be merged into existing tables in the database.

Establish claims database in analytics warehouse

Depending on the complexity of the existing database, it may be more prudent to create a separate database in the analytics data warehouse to store the new claims data. This separate database may also be used to store the Medicaid claims data that will be integrated from the planned MDL (Medicaid Data Lake) environment.

Perform data modeling for schema design of claims database

An analysis of the data sets that will be imported from the VHCURES database must be performed to inform the schema design of the new database. As part of the modeling process, the data fields from the various tables in the VHCURES database may be reorganized for optimum efficiency according to the expected analytics use cases.

Standardize claims data with respect to existing clinical data

Typical data fields such as date/time, phone numbers, zip codes, free text fields, etc. should be standardized across the two data sets for more efficient analytics. An analysis and data cleaning will need to be performed before the integrated data is useable.

Execute successful proof of concept for integration of claims data

Manual integration of an exported data set from VHCURES into VHIE will be performed at this stage. Validation that the data stores correctly and can be queried as per identified use cases will indicate a successful proof of concept.

Data Pipeline and Ingestion Processes

The following goals of the data pipeline and ingestion processes are

- Define data pipeline between VHCURES and VHIE
- Create, modify, or add ETL processes for claims data
- Automate pipeline components

Define data pipeline between VHCURES and VHIE

The infrastructure must be constructed to transport data between VHCURES and VHIE. The diagram for the proposed integrated data flow shows one way in which the current infrastructure is utilized, requiring the least modification and easiest setup. In this scenario, the existing S3 bucket in VHCURES is used to store data exports in csv format. That S3 bucket is exposed to the Rhapsody integration engine in VHIE which then ingests the files and processes the data according to its own ingestion process.

As this claims data is new, the components of the ingestion process must be created or modified for the new data file format and fields. As suggested by Medicasoft, the claims data may skip the FHIR

processing workflow and move right into the analytics data warehouse. The same would apply to the Medicaid claims data ingested from the MDL.

Another design consideration would be to use AWS Data Exchange to publish the claims data as a private data set and having VHIE subscribe to it. This creates a Redshift-to-Redshift ingestion process but skips much of the ETL process that is built in the Rhapsody data flow. Since that ETL process includes the MPI and terminology service, a new Redshift ETL process would need to be created which may negate some of the efficiency gained.

Create, modify, or add ETL processes for claims data

Introduction of claims data to the VHIE ingestion process necessitates modification to the existing process due to the new data source, data file format, and payload of the messages. Changes to the listener services and ETL processes would be necessary.

The default file format for the exported claims data in VHCURES has been noted to be csv, but there may be other formats that are more compatible with the current ingestion process in VHIE, such as JavaScript Object Notation (JSON) formatted to FHIR standard. Such a conversion would need to occur on the VHCURES side by a tool or script and may be possible to execute within the analytic enclave environment. This should only be considered if it is beneficial for the VHIE ingestion process and relieves some of the burden on VITL/Medicasoft to modify internal components for claims data.

Automate pipeline components

Once the new data pipeline is established, some of its components may be able to be automated to reduce operational burden. For example, the analytics enclave in VHCURES may be used by a data analyst to run a specific query and output the results to a file stored in S3. This process may be used to export the necessary data sets for VHIE, but it requires a manual step by a data analyst to execute. It should be feasible to create a script to generate this file and have it run via a scheduled job at the end of every month. In the same way, any manual steps involved in transporting the data should be automated as much as possible.

Data Migration

The goal of data migration is to move the historical data from VHCURES to VHIE. Once the claims data design in the data warehouse and data pipeline have been established, the historical claims data can be brought over to VHIE. Since this is not live operational data, it should be feasible to migrate this data in a single large batch. However, care should be taken to assess the total number of records and/or size of the data files that need to be brought over and the capacity of the systems to perform a bulk transfer as undoubtedly performance of the systems will be affected. A downtime may need to be scheduled if the affect is dramatic. It may be desirable to batch the data and transfer in multiple sessions during slow periods to reduce the affect to end users.

Risks and Challenges

As straightforward as the previous sections may seem, there are always challenges that present during plan execution. The following are some identified risks that may hamper or delay implementation.

SaaS to SaaS

Though VHCURES and VHIE are both hosted by the same cloud provider and use many of the same services and cloud components, they nevertheless are managed by their respective vendors. Available services that simplify functions such as data sharing, network access, user management, etc., within AWS, are not necessarily available to use between the two systems. Any solution that can be considered must only include functions that the respective vendor makes available.

Patient Identifiers

Currently, PHI data within the claims data files is hashed (encrypted) so that the patient/member is not identifiable[1]. The VHCURES data validation process when ingesting these files ensures that the PHI is properly hashed before accepting the file. However, the State of Vermont has updated the statute such that hashing of the PHI, along with the health plan patient identifier (a unique number assigned to the member), is no longer required. This will necessitate changes to the VHCURES data validation process at a minimum but will affect the MPI matching that currently exists with the hashed patient identifiers. Without modification, the historical claims data associated with a hashed identifier will not match new claims with a non-hashed identifier. A process will need to be created to update the hashed identifiers in the existing data to the non-hashed ones or to link them in some other manner. This would need to be rectified before integrating with VHIE.

Patient Matching

As noted, the claims data from VHCURES contains hashed PHI. Once integrated into VHIE, the claims data will not be able to match existing patients in the VHIE MPI since the demographics are not available. The anticipated analytics use cases may not require that the datasets be linked at the patient/member level. Nevertheless, this situation must be considered for future use cases.

Another key issue to consider, once the hashing is discontinued, is whether to migrate the historical claims data using the existing hashed patient identifiers or to update and reassign those to the actual patient identifiers. Also, a decision must be made where the correction will be made – in VHCURES before migrating to VHIE, or in VHIE after migrating. There are risks to either approach, with the possibility of affecting the historical data and existing reporting features of VHCURES in the former case, and risk of creating extraneous records in the analytics data warehouse in the latter. A separate project may need to be initiated to work solely on rectifying this situation.

In any case, a clear process will need to be defined for resolving patient matching issues such as duplicates, merges, and delinks. At this time, VITL indicates that the MPI can resolve a great number of these issues automatically, but this process and the tolerances associated with automatic resolution will need to be reviewed once the claims data is integrated due to the increased complexity. As the Verato platform uses machine learning (ML) to adapt its strategy for matching, it may be possible to train the MPI with the new data in order for it to correctly match the new claims data. This does require a data analyst or steward to manually match hundreds of records for the training process, so a method to match patients will still need to be established.

Due to the complexity of patient matching with the VHCURES claims data, Briljent recommends a federated governance team work specifically on this topic as an early item from the project backlog.

---

[1] Table 2 in the Appendix provides a list of the hashed fields.

Meaningful integrated data hinges on the ability to match the data sets through a common patient, so mitigation will be the highest priority.

Parallel Procurements and Projects

Other projects, particularly the MDWAS procurement, may compete with the completion of this project as they also involve updates to the VHIE environment and require the same resources. As such, implementation of updates MDWAS may necessitate changes to the implementation plan for VHIE VCURES integration. It will be necessary to have a means to collaborate with the Medicaid Data Warehouse and Analytics Service (MDWAS) team if implementation occurs concurrently.

Roadmap

Table 1 below provides a suggested timeline for execution of objectives discussed in the previous sections.

Table 1: Roadmap of activities by objective

Activity	Year	Month
Data Modeling, Design, and Standardization		
Identify data sets to be exported	Year 1	0-3
Establish claims database in analytics warehouse	Year 1	3-9
Perform data modeling for schema design of claims database	Year 1	6-9
Standardize claims data with respect to existing clinical data	Year 1	6-12
Execute successful proof-of-concept (PoC) for integration of claims data	Year 1	9-12
Data Pipeline and Ingestion Processes		
Define data pipeline between VHCURES and VHIE	Year 2	0-6
Create, modify, or add ETL processes for claims data	Year 2	3-6
Automate pipeline components	Year 2	6-9
Data Migration		
Migrate historical data from VHCURES to VHIE	Year 2	9-12

Next Steps

This integration plan outlines basic steps to integrate VHCURES claims data into the analytic warehouse in VHIE which contains clinical data. The plan takes the route of the quickest and easiest path to integration using existing components and processes in both systems. Some alternative paths are mentioned in the plan, but a more detailed solutioning discussion should occur with the enterprise

architect and project owners to ensure that the plan aligns with the overall goals of the State and the future vision of the Unified Health Data Space.

Furthermore, additional technical discussions should be had with VITL/Medicasoft and Onpoint Health Data to explore any other possibilities that may improve the integration plan such as any new features to be released or existing modules that are not currently in use.

Finally, a planning discussion should occur with the MDWAS project team once the procurement is completed to collaborate efforts in case there are overlapping or competing requirements for VHIE resources. Execution of this integration plan may affect the MDWAS integration plan and vice versa. In this way, the State will be able to complete both projects as smoothly as possible.

Appendix

For reference, Table 2 is a screen shot of the VHCURES data dictionary with the hashed fields highlighted.

Table 2: VHCURES data dictionary with hashed fields highlighted

ID	Warehouse Name	Common Name	Type	Length
MEM-CORE001	member_id	Member ID	Number	38
MEM-CORE002	extract_id	Extract ID	Number	38
MEM-CORE003	internal_member_id	Internal Member ID	Number	38
MEM-CORE004	last_name	Last Name (Encrypted)	Varchar2	272
MEM-CORE005	first_name	First Name (Encrypted)	Varchar2	272
MEM-CORE006	middle_name	Middle Name or Initial (Encrypted)	Varchar2	272
MEM-CORE007	city	Member City	Varchar2	100
MEM-CORE008	state	Member State or Province	Varchar2	2
MEM-CORE009	zip	Member ZIP/Postal Code	Varchar2	5
MEM-CORE010	birth_dt	Member's Date of Birth (Encrypted)	Varchar2	272
MEM-CORE011	death_dt	Member's Date of Death (Encrypted)	Varchar2	272
MEM-CORE012	ssn	Social Security Number (Encrypted)	Varchar2	272
MEM-CORE013	gender_code	Gender Code	Varchar2	2
MEM-CORE014	external_member_identifier	Unique Member Identifier (Submitter Supplied)	Varchar2	50
MEM-CORE015	member_flag	Member Flag	Varchar2	1
MEM-CORE016	subscriber_member_id	Subscriber Member ID	Number	38
MEM-CORE017	subscriber_internal_member_id	Subscriber Internal Member ID	Number	38
MEM-CORE018	contract	Plan-Specific Contract Number (Encrypted)	Varchar2	272
MEM-CORE019	member_sequence	Member Suffix or Sequence Number	Varchar2	20
MEM-CORE020	hispanic_id	Hispanic Indicator ID	Number	38
MEM-CORE021	hispanic_code	Hispanic Indicator Code	Varchar2	20
MEM-CORE022	race_id1	Race Code (1) ID	Number	38
MEM-CORE023	race_code1	Race Code (1)	Varchar2	20
MEM-CORE024	race_id2	Race Code (2) ID	Number	38
MEM-CORE025	race_code2	Race Code (2)	Varchar2	20
MEM-CORE026	other_race	Race (Other) Description	Varchar2	100
MEM-CORE027	ethnicity_id1	Ethnicity Code (1) ID	Number	38
MEM-CORE028	ethnicity_code1	Ethnicity Code (1)	Varchar2	20
MEM-CORE029	ethnicity_id2	Ethnicity Code (2) ID	Number	38
MEM-CORE030	ethnicity_code2	Ethnicity Code (2)	Varchar2	20
MEM-CORE031	other_ethnicity	Ethnicity (Other) Description	Varchar2	100