



**Requirement Specification for
Member
Part of the Open EVV Series of Interfaces**

Version 7.5

Sandata Technologies, LLC
26 Harbor Park Dr.
Port Washington, NY 11050
Toll Free: 800-544-7263
Tel: 516-484-4400
Fax: 516-484-0679
Email: info@sandata.com
Web: www.sandata.com



This document and the information contained herein are confidential to and the property of Sandata Technologies, LLC. Unauthorized access, copying and replication are prohibited. This document must not be copied in whole or part by any means, without the written authorization of Sandata Technologies, LLC. This document should be used only for intended purpose only.



Table of Contents

1 Overview	5
1.1 Intended Audience	5
1.2 Data Type Format Details	5
2 Member Interface	6
3 Data Exchange	7
3.1 Real-Time Transactions	7
3.2 Delimiter Separated Values (DSV)	13
4 Appendixes	27
4.1 Assumptions	27
4.2 Other Important Points to Note	27
4.3 Legend	27
4.4 Acronyms and Definitions	28
4.5 Time Zone List	28
5 References	30
5.1 DSV	30
5.2 End of Line (EOL)	30
5.3 Error Detection	30
5.4 UTF-8	30
5.5 OpenPGP	30



Revision History

Version	Description	Date Updated
7.1	Removed ClientID from XREF segment as it is unnecessary	4/2/2019
7.2	Update required fields in XREF segment. Remove account from XREF segment as it is unnecessary.	4/12/2019
7.3	Removed references to consuming "Full" files – leaving incremental as only option.	5/20/2019
7.4	Updated ClientWorkerXREF JSON Sample	7/29/2019
7.5	Broke Client Designees out into their own segment, supporting multiple designees per member. Added client designee relationship field. Added client enrollment date field.	8/12/2019

1 Overview

This specification is intended to document the requirements for using the Sandata Real Time Interface (part of the Open EVV Series of interfaces) for receiving client / member information from 3rd party systems including Payers and MCOs.

A companion guide will be created for each Payer / Program implemented to specify agreed upon frequencies, additional required fields and those fields which will be omitted or left to the sender’s discretion.

1.1 Intended Audience

The intended audience of this document is:

- Project Management and Technical teams at Sandata.
- Project Management and Technical teams at designated Payers/MCOs/Vendors who will be implementing this interface.

1.2 Data Type Format Details

Data Type	Description	Example
DateTime	<p>The date and time is represented as a string with the following format: YYYY-MM-DDTHH:MM:SSZ</p> <p>All times will be provided in UTC.</p> <p>If time is not material, it will be provided as is expected.</p>	2016-12-20T16:10:28Z
Date (only Date)	<p>The data is represented as a string with the following format: YYYY-MM-DD</p> <p>Date only will be sent in UTC format.</p>	2016-12-20

Data Type	Description	Example
Timezone	<p>All time for tracking visits will be in UTC.</p> <p>All time zone values will be derived from the Internet Assigned Numbers Authority (IANA) Time Zone Database, which contains data that represents the history of local time for locations around the globe. It is updated periodically to reflect changes made by political bodies to time zone boundaries, UTC offsets, and daylight-saving rules.</p> <p>The Time zone name expected in each transaction is the actual Time zone where the event took place. i.e. US/Eastern</p>	<p>A complete list of time zones can be found at: https://www.iana.org/time-zones</p> <p>See Appendix for list of Timezones</p>
String	<p>A string is a row of zero or more characters which can include letters, numbers, or other types of characters as a unit, not an array of single characters. (e.g. plain text).</p>	<p>“This is a string” (See Wikipedia String)</p>
Integer	<p>An integer is a numeric value without a decimal. Integers are whole numbers and can be positive or negative.</p>	<p>52110 (positive) -87721 (negative) (See Wikipedia Integer)</p>
Decimal	<p>A floating point number is referred to as a decimal. Can be positive or negative.</p>	<p>8221.231 (positive) -71.214 (negative) (See Wikipedia Decimal)</p>
Boolean	<p>A logic predicate indicator that can be either true or false.</p>	<p>True False See Wikipedia Boolean</p>

2 Member Interface

This specification is intended for a sender to provide member eligibility information including basic member demographic data. This import may also be referred to as Client, Member, Recipient, Individual or Patient. Member information will be made available to the appropriate provider(s) based on the information within the file or based on authorization(s) received.



3 Data Exchange

Sandata supports data exchange via two mechanisms: a real-time, RESTful API and flat-file processing (DSV).

While Sandata supports both mechanisms, the default is a RESTful API with JSON to greatly reduce complexity of customized implementations.

3.1 Real-Time Transactions

Members may be sent via a real-time RESTful API for processing. Sandata will take each request as it is received, process the member and return a response

Sandata will provide real-time RESTful API endpoints for the customers in a UAT environment for user acceptance testing as well as production. This document contains the technical details for utilizing this API. The API is designed to be a service-oriented architecture (SOA). All transactions will utilize the JSON format which is the JAVA equivalent to XML. JSON, like XML is self-describing. A WADL (equivalent to the WSDL) will be provided using the [API documentation provided by Swagger](#).

Payers must be able to send and consume member data and responses in a **JSON** format. **JSON** allows multiple 'child' entities for a parent (See JSON Request Example)

NOTE: For testing purposes, generic, de-identified files will be provided, or data for testing will be identified by the payer based on available or constructed data. Testing these files will be part of the overall system testing process. Mutually agreed upon dates will be determined for joint testing and included in the overall project plan.

3.1.1 Representational State Transfer (REST) Interface

Sandata has developed a RESTful interface that allows for a client to send data as real time transactions with appropriate responses rather than in batches of text files for periodic processing

In a Sandata RESTful web service, requests made to a resource's URL will elicit a response with a payload formatted in JSON. The response can confirm that some alteration has been made to the stored resource, and the response will provide any errors that may have occurred. When HTTP is used for processing members, you will only need to execute a POST HTTPS request method.

3.1.2 HTTPS (TLS 1.2)

Sandata's RESTful interfaces support TLS v1.2 (a successor to SSL) which provides a layer of security and reliability by exchanging PHI information as encrypted data packets between Sandata and other payer systems.



3.1.3 Members Real-Time Processing

The members real-time processing interfaces refers to Sandata's RESTful HTTPS endpoints for receiving members. Sandata will provide the Payer with URL endpoints for UAT and Production. Sandata will also provide the Payer with a username and password that will use Basic Authentication to validate the request. The Payer will receive a 401 HTTP error code if the username and/or password does not match.

3.1.4 JSON Examples

Request Payload Example

Below, find a sample payload (body) that could be sent to the Sandata real-time RESTful API. See the table in section 3.2.16 for a detailed description of each field.

```
{
  "ClientID": "55267687",
  "ClientFirstName": "John",
  "ClientMiddleInitial": "D",
  "ClientLastName": "Doe",
  "ClientSSN": "123456789",
  "ClientMedicalRecordNumber": null,
  "ClientCustomID": "123456789",
  "ClientOtherID": null,
  "ClientSuffix": null,
  "Action": "A",
  "ProviderQualifier": "SandataID",
  "ProviderID": "123456",
  "CaseManager": null,
  "ClientCaseManagerEmail": null,
  "ClientCoordinatorEmail": null,
  "ClientLanguage": null,
```



```
"ClientGender": "M",  
"ClientBirthDate": "1990 - 01 - 02",  
"ClientMaritalStatus": "M",  
"ClientEmail": "test@sandata.com",  
"ClientPriority": null,  
"ClientTimeZone": "US/Eastern",  
"ClientEnrollmentDate": "2017-07-01",  
"ClientDesignee": [{  
    "ClientDesigneeFirstName": null,  
    "ClientDesigneeLastName": null,  
    "ClientDesigneeEmail": null,  
    "ClientDesigneeStatus": "04",  
    "ClientDesigneeStartDate": null,  
    "ClientDesigneeEndDate": null,  
    "ClientDesigneeRelationship": null  
}],  
"ClientEligibility": [{  
    "PayerID": "1890017",  
    "PayerProgram": "program",  
    "PayerService": "T1001",  
    "PayerRegion": "NA",  
    "ClientEligibilityDateBegin": "2017-12-12",  
    "ClientEligibilityDateEnd": "2018-12-12",  
    "ClientStartOfCareDate": "2017-12-12",  
    "ClientEndOfCareDate": "2018-12-12",
```

```
"ClientPrimaryDiagnosisCode": "12345 ",
"ClientSecondaryDiagnosisCode ": null,
"ClientStatus ": "02",
"ClientStatusDate": "2018-01-01",
"Modifier1": "A1",
"Modifier2": "A2",
"Modifier3": "A3",
"Modifier4": "A4"
}},
"ClientContact": [{
  "ClientContactType": "Other",
  "ClientContactFirstName": "Jane",
  "ClientContactLastName": "Done",
  "ClientContactPhoneType": null,
  "ClientContactPhone": null,
  "ClientContactEmailAddress": "jane@sandata.com",
  "ClientContactAddressLine1": "998 Little Place Drive",
  "ClientContactAddressLine2": null,
  "ClientContactCity": "Manhattan",
  "ClientContactState": "NY",
  "ClientContactZip": "10017"
}],
"ClientAddress": [{
  "ClientAddressType": "Home",
  "ClientAddressLine1": "777 East 7th Street",
```

```
        "ClientAddressLine2": null,  
        "ClientCounty": "HA",  
        "ClientState": "NY",  
        "ClientZip": "11235"  
    },  
    "ClientPhone": [{  
        "ClientPhoneType": "Home",  
        "ClientPhone": "7185551212"  
    }, {  
        "ClientPhoneType": "Business",  
        "ClientPhone": "7185551214"  
    }],  
    "ClientWorkerXref": [{  
        "VendorCode": "12V",  
        "EmployeePIN": "999999999",  
        "Service": "T1001",  
        "XrefStartDate": "2018-01-01",  
        "XrefEndDate": "2019-12-31"  
    }]  
}
```

Response Payload Examples



NOTE: The response example shows the payload (body) that will be a response from the Sandata real-time RESTful API. The response is contained as part of the “data” entity which is part of the standard Sandata HttpResponse entity. This response may be augmented over time to contain additional information. Consumers of the API should be able to handle responses that contain additional data elements.

id – This field is a RESTful service transaction globally unique ID (GUID) which is generated by Sandata. Please log this GUID as it will help Sandata Tier3 support and troubleshoot any issues.

status – This status has two possible values:

- **SUCCESS:** Indicates that the request was received and processed successfully by the Sandata backend.
- **FAILED:** Indicates that there was some error detected by the Sandata backend. E.g. 500 Server Error

NOTE: Both of these states are returned with an HTTP 200 response code.

messageSummary – This field This field will contain either null for status=SUCCESS or “Parameter Error” for status=FAILED. This would typically occur for a “POST” without BODY.

messageDetail – This field will contain either null for status=SUCCESS or a detailed service error message for status=FAILED. E.g. “Database Unavailable”

failedCount – the number of items in the request that resulted in some error

succeededCount – the number of items in the request that ended in a successful result

data – This entity will contain details of the JSON response. Examples can be provided upon request.

Successful Response Example

```
{
  "id": "d25cbb0c-2043-4a71-ae7c-8e917b71096c",
  "status": "SUCCESS",
  "messageSummary": null,
  "messageDetail": null,
  "failedCount": 0,
  "succeededCount": 2,
  "data": [{
```

```
"ProviderID": "12345",  
"ClientID": "98765",  
"ClientCustomID": "23456",  
"ClientOtherID": "78901",  
"status": "SUCCESS",  
"messageSummary": null,  
"messageDetail": null  
  }  
}
```

Failed Response Example – This example is caused by batch-level failure including file/transmission corruption or incorrect JSON.

```
{  
  "id": "228cb2fa-50da-453e-b9a7-7f35da47c492",  
  "status": "FAILED",  
  "messageSummary": "Request Failed",  
  "messageDetail": "Your request has been received and logged successfully. However, an internal error was triggered. The Sandata technical team has been notified. Please retry your request. If you continue to experience this error, contact Sandata and provide the GUID [228cb2fa-50da-453e-b9a7-7f35da47c492] for the failed transaction."  
}
```

3.2 Delimiter Separated Values (DSV)

A delimited file is a text file used to store data, in which each line represents a single record (i.e. Provider) and each line has fields separated by the agreed upon delimiter. Compared to a fixed-length formatted files that uses

spaces or other filler characters to force the length of a given field to be fixed in width/size for every value, a delimited file has the advantage of allowing field values of any length. Additionally, when accompanied by a “header row” (the first row in a file) that provides for the names of each column of data, columns of data can arrive in any order and columns may be added or removed without having to re-write rules for data transformation.

NOTE: The very first line within the DSV is the header record. (See [Header Record](#))

3.2.1 Supported Delimiters

Acceptable delimiters supported by this specification include:

- Pipe or Vertical Bar (|); ASCII 124 or UTF-8 007C
- Comma (,); ASCII 44 or UTF-8 002C

3.2.2 End of Line Characters

Each record within the Member DSV will be located on a new line, which is composed of two characters, carriage return (\r) and line feed (\n).

3.2.3 Double Quotes

- Each field will be enclosed with double quotes (“”).
- However, NULL data does NOT use double quotes

Example: “<PAYER>”|”HHS”|”North America”|”MedicaidID”|”123456789”

3.2.4 Character Encoding

Each field within the DSV file must conform to the [ASCII/UTF-8](#) character encoding standard.

3.2.5 Header Record

The header record provides for the names of each column of data found in the DSV. Columns of data can arrive in any order and columns may be added or removed without having to re-write rules for data transformation.

NOTE: Rules around columns data points will be discussed with Sandata during implementation. Removing columns from the DSV that are critical to the import process will cause an error and the entire file will be rejected.

- The header record is the first record at the top of the delimited file.
- The header record is *required*.
- The field names in the header record, also known as column names, must conform to the names provided by Sandata. (See Member DSV Field Names)
- Customers, at their discretion, may exclude *non-required* fields.

Example: "ClientID"|"ClientFirstName"|"ClientMiddleInitial" | ... | "XRefEndDate"

3.2.6 File Naming Convention

The file naming convention is important to help with validation, entity mapping, dates and times to make sure files are not overwritten and are loaded in the order they are received, extensions to drive the parsing and decryption logic, etc.

NOTE: Use underscores (_) to separate each variable section of the file name.

[Prefix]_[EntityName]_[YYYYMMDD]_[HHMMSS.SSS]_[Incremental].[FileExtensions]

[Prefix] is a customer specific string agreed upon with Sandata during implementation. The file prefix must be included with all files provided by the customer ("**<SENDER>_EVV**")

[EntityName] is the name of the domain specific name of the parent entity that reflects the data fields within the DSV file ("Member")

[YYYYMMDD] is the four-digit year, two-digit month and two-digit day that the file was created

[HHMMSS.SSS] is the two-digit hours, two-digit minutes, two-digit seconds, and three-digit milliseconds values (Military Time)

- [HHMMSS.SSS] file value can be optional if we are consuming a daily file

[Incremental]

[Incremental] signifies that the file contains only new and/or updated data from the source system

[Incremental] file value can be optional if we are consuming a daily file

[FileExtensions]

[.csv] signifies a comma separated file

[.dsv] signifies a delimiter separated file (specific delimiters are agreed upon with the customer during implementation)

[.zip/.gzip/.gz/.tar/.7z] signifies the compression used

[.gpg] signifies that the file has been encrypted with PGP [See [File Encryption](#)]

Example Format

<SENDER>_EVV_Member_20180817.dsv.gpg

3.2.7 File Encryption

File encryption is encouraged to add an additional layer of security for sensitive PHI data. Files are processed over Secure FTP (SFTP) which provides its own layer of encryption as well.

- Sandata supports file encryption using OpenPGP ([RFC4880](#)).
- Sandata will provide customers with a public key upon implementation.
- PGP encrypted files will append the "gpg" file extension.

3.2.8 Cryptographic Hash (Optional)

A cryptographic hash function can provide strong assurance about data integrity, whether changes to the data are accidental (e.g., due to transmission errors) or maliciously introduced. Any modification to the data will be detected through a mismatching hash value. Furthermore, given some hash value, it is infeasible to find some input data (other than the one given) that will yield the same hash value.

- The customer can calculate the hash value for each DSV file and provide that value in the control file.
- When calculating the hash, the customer can use any of the following hash functions:
 - SHA-1
 - SHA-2 (SHA-256/512)
 - SHA-3 (Most Secure) (Recommended)
- NOTE: MD5 is no longer supported as it has known security vulnerabilities
 - This hash value of a file is optional. Sandata will validate the hash if one is provided in the control file under the “Hash” column. (See Control File)

3.2.9 Control Files

Control files are used as a quality control mechanism to ensure file integrity following transmission.

- The customer will provide Sandata with an outbound control file.
- Sandata will provide the customer with an inbound control file.
- The control file will be named as follows
 - [Prefix]_[Direction]_ControlFile_[YYYYMMDD]_[HHMMSS.SSS].[FileExtensions]
 - [Prefix] is a customer specific string agreed upon with Sandata during implementation
 - [Direction]
 - Outbound – Customer to Sandata
 - Inbound – Sandata to Customer
 - [YYYYMMDD] is the four-digit year, two-digit month and two-digit day that the file was created
 - [HHMMSS.SSS] is the two-digit hours, two-digit minutes, two-digit seconds, and three-digit milliseconds values (Military Time)
 - [HHMMSS.SSS] file value can be optional if we are consuming a daily file
 - [FileExtensions]
 - [.csv] signifies a comma separated file
 - [.dsv] signifies a delimiter separated file (specific delimiters are agreed upon with the customer during implementation)
 - [.zip/.gzip/.gz/.tar/.7z] signifies the compression used
- The control file will be a DSV file using the same delimiter agreed upon with the customer during implementation
- The outbound control file will have the following column names for the header row (assuming pipe (|) delimiter value). Quotation marks are optional in control file.

- "FileName"|"RecordCount"|"StartDateTime"|"EndDateTime"|"Hash"
 - FileName: (See **File Naming Convention**)
 - RecordCount: Total number of records found in the DSV (not including the header row)
 - StartDateTime: The start date and military time when the records in the DSV were queried from. (See **Date Time Format**) [Optional]
 - EndDateTime: The end date and military time when the records in the DSV were queried from. (See **Date Time Format**) [Optional]
 - Hash: Cryptographic hash value generated by the given file. (See **Cryptographic Hash**) [Optional]
- The inbound control file will have the following column names for the header row (assuming pipe (|) delimiter value)
 - "FileName"|"RecordCount"|"StartDateTime"|"EndDateTime"|"Hash"|"Success Count"|"Failed Count"
 - FileName: (See **File Naming Convention**)
 - RecordCount: Total number of records found in the DSV (not including the header row)
 - StartDateTime: The start date and military time when the records in the DSV were queried from. (See **Date Time Format**)
 - EndDateTime: The end date and military time when the records in the DSV were queried from. (See **Date Time Format**)
 - Hash: Cryptographic hash value generated by the given file (See **Cryptographic Hash**)
 - Success Count: Total records that were processed successfully
 - Failed Count: Total records that were not processed successfully
- Example outbound control file:
 - <sender>_EVV_Outbound_ControlFile_20180817.dsv.gpg
 - "FileName"|"RecordCount"
 - "SENDER_EVV_Provider_20181002.dsv"|"2012"
 - "SENDER_EVV_Member_20181002.dsv"|"12"
 - "SENDER_EVV_PriorAuth_20181002.dsv"|"22"
 - "SENDER_EVV_Outbound_ControlFile_20181002.dsv"|"5"
 - "2018-07-31T20:00:00Z"|"2018-10-30T16:00:00Z"

The last row of the control file is a date and time range of the extracts, for informational purpose only, and would only be used by SENDER for possible future use in regeneration efforts. Not expected to be validated by Sandata.

- Example inbound control file:
 - <SENDER>_EVV_Inbound_ControlFile_20180817.dsv.gpg



"FileName"|"RecordCount"|"StartDateTime"|"EndDateTime"|"Hash"|"Success Count"|"Failed Count"

"<SENDER>_EVV_Member_Errors_20180817.dsv.gpg"|"2012"|"2018-09-18T00:00:00Z"|"2018-09-18T23:59:59Z"|"cjpqr032alimp883jasddejkm"|"2012"|"0"

3.2.10 Error File

ERROR HANDLING PROCESS

Sandata will notify Sender via email to alert of any errors found in processing each file that was imported.

Sandata will not send emails or error files if there are no errors detected for the delivery.

The email would be addressed to EVV Interface_SENDER@SENDER_DOMAIN.com

The email Subject would include "SENDER-Sandata errors: {date of files (probably same for all)}"

The email Body would include (at a minimum) lines for "file name", "number of errors found"

Sandata will provide the customer with an error file for each file that was imported.

Only those records that caused error would be sent in the error file.

The error file will add an "Error Description" column to the end of record.

"Error Description": This is a string value describing the error and/or errors that were encountered when trying to process the record

The naming of the error file is the same as the naming pattern of the source file (See File Naming Convention) with an "Error" label appended to the [Entity]

Example

SENDER_EVV_Member_Error_20180817.dsv.gpg

3.2.11 File Transport

Files will be consumed and delivered via Secure FTP (SFTP). The target SFTP server will be hosted by Sender. The host IP, username, password and *optional* public cryptographic key have been discussed and tested during implementation.

3.2.12 File Location

DSV files will be located on the secure SFTP server, in folder created specifically for Sandata,

"/Prod/From_SENDER".

3.2.13 File Frequency

Sandata will accept files on a daily schedule. The initial run for the Member file will contain all members that could have authorizations. The ongoing daily job will run and deliver files at about 2AM, Monday through Friday.

3.2.14 Member File Logic



- For an authorization to be applied to a provider, the member information must be received at least once. If there is no matching member, the authorization will be held until the member is received.
- Sandata is expecting to receive only those members who are active in the program and if authorizations are in use, the member must have an active authorization. Once a member is transmitted to Sandata, the member must only be resent if there are changes to be applied to the prior information.

3.2.15 Member DSV Field Names

- The fields listed below are the fields available for transmission to Sandata EVV.
- Required columns must have data, otherwise the system will reject the record.
- The file will be rejected if a header column name is unknown to the Sandata system.
- If a field is not required, it does not need to be included.

3.2.16 Parent-Child Relationships

- The client data represented in the DSV file naturally has some parent-child (one-to-many) relationships
- These can be referenced in the flat file through the use of the “SegmentName” field
- Valid segment names are:
 - ClientDesignee
 - ClientEligibility
 - ClientContact
 - ClientAddress
 - ClientPhone
 - ClientWorkerXref
- If the DSV row represents basic client data, it is expected the “SegmentName” will be blank and no data for any of the above segments will be provided
- If the DSV row represents any of the above segments, it is expected that the “SegmentName” is populated and the only data in the row is the basic client data and the specified segment
- If segments beyond basic client information are provided, there are specific fields that must be included for each. See the table below for a list of these required fields per segment. Note the segment descriptions below mention some scenarios where the segments themselves would be required.
- Since all data will be transmitted in a single file, columns for each row must be consistent and match the header. For instance, even if a segment only includes a handful of columns, empty values must be included in all other columns for the row. Each row should exactly match the number of columns provided in the header record.

Index	Column Name	Description	Max Length	Type	Required
Basic client information – biographical and demographic information. This information should be consistent for every row for a client that may exist in the file (due to multiple segments).					
1	ClientID	Sandata Assigned ClientID. This value can be automatically assigned by Sandata EVV. Note that this value may be used as the client identifier for telephony and MVV when Client ID entry is applicable. This value must uniquely identify the client in the Sandata system. This attribute is only required when the customer is supplying this value to Sandata.	10	String	
2	ClientFirstName	Client's First Name.	30	String	Yes
3	ClientMiddleInitial	Client's Middle Initial. May be required if needed for billing.	1	String	
4	ClientLastName	Client's Last Name.	30	String	Yes
5	ClientSSN	Client's Social Security Number. If the Field is left empty, ClientOtherID must be populated. Not required if ClientOtherID sent. Numbers only, no dashes and leading zeroes must be included. May be required if needed for billing. Format - #####.	9	String	
6	ClientMedicalRecordNum	Client's medical record number if it is applicable.	12	String	
7	ClientCustomID	This is mapped to the member's Medicaid_ID	24	String	Yes
8	ClientOtherID	Additional Client User-Defined ID	24	String	Yes
9	ClientSuffix	Client Suffix (eg. Sr, Jr, III, IV, V (no special characters)).	4	String	
10	Action	Values: A, C, D. Is this a new record (Add - A), a change to an existing record (Change - C) or a Deletion (D). If (D) is provided the record will be considered inactive as of midnight the date prior to the receipt date. Note that a client cannot be deleted within the Sandata system.	1	String	Yes
11	ProviderQualifier	Identifier being sent as the unique identifier for the provider. Values: SandataID, NPI, API, MedicaidID, TaxID, Taxonomy, Legacy, Other. Required if client is being directly associated with a provider versus being associated through the authorization or a cross reference file.	20	String	

Index	Column Name	Description	Max Length	Type	Required
12	ProviderID	ID for the provider. ID type identified by ProviderQualifier. Required if client is being directly associated with a provider versus being associated through the authorization or a cross reference file.	64	String	Conditional
13	CaseManager	Case Managers are individuals who coordinate all aspects of the care of individual patients.	25	String	
14	ClientCaseManagerEmail	Email address of the Payer case manager. Used for setup for alerts to be received by a case manager. An email group can also be supplied if this alert is intended for more than one recipient.	64	String	
15	ClientCoordinatorEmail	Email address of client's agency coordinator.	50	String	
16	ClientLanguage	Client's language. The list of acceptable values will be determined during implementation.	32	String	
17	ClientGender	Client's Gender. Values: O=Unknown or Other, M=Male, F=Female.	1	String	
18	ClientMaritalStatus	Client's Marital Status. Values: M = Married, S = Single, W = Widowed, O = Other.	1	String	
19	ClientBirthDate	Client's Date of Birth. Required for billing. Format YYYY-MM-DD (zero filled). e.g. 2018-06-01	10	Date	
20	ClientEmail	Client's email address. Required for client portal access.	50	String	
21	ClientPriority	Allows designation of a client's priority. Generally used to designate clients whose service is critical. Values will be determined during implementation if applicable.	2	String	
22	ClientTimeZone	Client's primary time zone. Depending on the program, this value may be defaulted or automatically calculated. Please see the appendix for acceptable values.	64	String	
23	ClientEnrollmentDate	Client's initial enrollment date.	10	Date	
24	SegmentName	If this is a child segment, the name of the segment. Valid values are found above. NOTE: Field should be omitted when using the REST API.	30	String	
Segment – ClientDesignee : Designee(s) for the associated client. A given client can be associated with more than one designee. The segment is not required, but if provided, the required fields must be sent.					
25	ClientDesigneeFirstName	First Name of the Client Designee.	30	String	Yes

Index	Column Name	Description	Max Length	Type	Required
26	ClientDesigneeLastName	Last Name of the Client Designee.	30	String	Yes
27	ClientDesigneeEmail	Email address of the Client Designee.	50	String	Yes
28	ClientDesigneeStatus	<p>Status of the Client Designee pertaining to Sandata system access. If the ClientDesigneeStatus is sent, ClientDesigneeStartDate and ClientDesigneeEndDate are not required.</p> <p>(Provide the 2-digit code including the 0)</p> <p>Sandata System can either populate the start or end date based on the date of receipt of the status or the source system can send the activation and termination date.</p> <p>(Please note Activation and termination dates cannot be backdated or future dated)</p> <p>Available Values: 02 = Active, 04 = Inactive.</p>	2	String	Conditional – Required when ClientDesigneeStartDate and ClientDesigneeEndDate are not provided
29	ClientDesigneeStartDate	<p>The date Client Designee was assigned. Future date is not acceptable. If the ClientDesigneeStartDate is sent, ClientDesigneeStatus is not required.</p> <p>Format YYYY-MM-DD</p>	10	Date	Conditional – Required when ClientDesigneeStatus is not provided
30	ClientDesigneeEndDate	<p>The date Client Designee was terminated. Future date and Back date is not acceptable. If the ClientDesigneeEndDate is sent, ClientDesigneeStatus is not required.</p> <p>Format YYYY-MM-DD</p>	10	Date	Conditional – Required when ClientDesigneeStatus is not provided
31	ClientDesigneeRelationship	Relationship of the Designee to the Client	30	String	
<p>Segment – ClientEligibility : Payer – Program – Service for the associated client. A given client can be associated with more than one payer and within each payer one or more programs and services. PayerProgram and PayerService fields are required only if the Sender is not providing Sandata with the Authorization file.</p>					
32	PayerID	<p>Sandata EVV assigned ID for the payer. Required if the file is being supplied by a payer.</p> <p>Determined during the implementation process.</p> <p>At least 1 payer is required for each client.</p>	64	String	Yes

Index	Column Name	Description	Max Length	Type	Required
33	PayerProgram	If applicable, the program to which this client belongs. List of values to be determined during implementation.	9	String	Conditional
34	PayerService	If applicable, the service to which this client belongs. List of values to be determined during implementation. Generally provided as a HCPCS code. Field may be larger is a different value is selected.	5	String	Conditional
35	PayerRegion	If applicable, the region in which this client is being provided services. List of values to be determined during implementation.	2	String	
36	ClientEligibilityDateBegin	Client Eligibility Begin Date. Format YYYY-MM-DD.	10	Date	Yes
37	ClientEligibilityDateEnd	Client Eligibility End Date. Format YYYY-MM-DD.	10	Date	
38	ClientStartOfCareDate	Start of Care Date. Format YYYY-MM-DD.	10	Date	
39	ClientEndOfCareDate	End of Care Date. Format YYYY-MM-DD.	10	Date	
40	ClientPrimaryDiagnosisCode	The client's primary diagnosis code in ICD-10 format.	10	String	
41	ClientSecondaryDiagnosisCode	The client's secondary diagnosis code in ICD-10 format.	10	String	
42	ClientStatus	The client's current status. Provide the 2-digit code including the 0. Available values: 01 = Pending, 02 = Active, 03 = Hold, 04 = Inactive.	2	String	Yes
43	ClientStatusDate	The date of the last status change. If not provided, current date will be assumed. Entered by Payer. Format YYYY-MM-DD.	10	Date	
44	Modifier 1	First modifier if applicable. May include modifier information to be used for service provision and billing.	2	String	
45	Modifier 2	Second modifier if applicable. May include modifier information to be used for service provision and billing.	2	String	
46	Modifier 3	Third modifier if applicable. May include modifier information to be used for service provision and billing.	2	String	

Index	Column Name	Description	Max Length	Type	Required
47	Modifier 4	Fourth modifier if applicable. May include modifier information to be used for service provision and billing.	2	String	
Segment – ClientContact : Client Authorized Representative / Emergency Contact					
48	ClientContactType	Client Contact Type. Values: Family, Other.	12	String	
49	ClientContactFirstName	Client Contact First Name. Entered by provider agency.	30	String	
50	ClientContactLastName	Client Contact Last Name. Entered by provider agency.	30	String	
51	ClientContactPhoneType	Client Contact's Phone Type. Values: Business, Home, Mobile and Other.	12	String	
52	ClientContactPhone	Client Contact Home Phone Number. Entered by provider agency. Format #####.	10	String	
53	ClientContactEmailAddress	Client Contact's email address. Required if this client will be authorized to login to the client portal as the client's authorized representative and approve timesheets on behalf of the client.	64	String	
54	ClientContactAddressLine1	Client Contact's Street Address, Line 1.	30	String	
55	ClientContactAddressLine2	Client Contact's Street Address, Line 2.	30	String	
56	ClientContactCity	Client Contact's City.	30	String	
57	ClientContactState	Client Contact's State. Two Character standard abbreviation.	2	String	
58	ClientContactZip	Client Contact's Zip Code. 9-digit primary address zip code. If additional 4 digits are not known, or not provided, Sandata will provide zeros (trailing). Format #####. If ZipCode is sent as 5 digits (assuming leading zero is present for appropriate zipcodes (e.g. Augusta, ME 04330)), then trailing zeros will be added.	9	String	
Segment – ClientAddress : – All addresses where service could be provided should be supplied. If mobile devices are in use, at least 1 address should be supplied. Note that P.O. Boxes cannot be used for Mobile verification purposes. Sandata offers the option for the Payer to provide one address (considered primary) and the provider to provide optional alternate addresses.					
59	ClientAddressType	Values: Home, Business, Other. Note that multiple of the same type can be provided.	12	String	Yes
60	ClientAddressLine1	Street Address Line 1 associated with this address. PO Box may not acceptable for Billing and PO Box will not function correctly for MVV.	30	String	Yes



Index	Column Name	Description	Max Length	Type	Required
61	ClientAddressLine2	Street Address Line 2 associated with this address.	30	String	
62	ClientCounty	County associated with this address.	25	String	
63	ClientCity	City associated with this address.	30	String	Yes
64	ClientState	State associated with this address. Two Character standard abbreviation.	2	String	Yes
65	ClientZip	Zip Code associated with this address. Required for Billing. 9-digit primary address zip code. If additional 4 digits are not known, or not provided, Sandata will provide zeros (trailing). Format #####. If ZipCode is sent as 5 digits (assuming leading zero is present for appropriate zipcodes (e.g. Augusta, ME 04330)), then trailing zeros will be added.	9	String	Yes
Segment – ClientPhone : – All phone numbers where service could be provided should be supplied.					
66	ClientPhoneType	Values: Home, Mobile, Business and Other. Note that multiple of the same type can be provided. <u>If this value is not provided, Sandata will automatically assign the value of “Other”.</u>	12	String	
67	ClientPhone	Client phone number. Format #####.	10	String	Yes
Segment – ClientWorkerXref : Cross Reference (Client/Employee) -This element needs to be utilized for clients provided services under a consumer directed program or for a program where authorizations cannot be provided where the employees and their association to the client(s) are known by the Payer. Additional elements may be added for a particular program and can be found in the program companion guide.					
68	VendorCode	Created at the time of account creation. This number is generated by Sandata EVV as the Provider Account Number. Client may send this attribute but it will not be validated or stored in Sandata’s DB. This field will not be included in the DW Export as well.	10	String	
69	EmployeePIN	Unique identifier used by the employee when calling into the Santrax EVV system. The system will perform optimally if this value is the same length for all employees and should be no less than 4 digits.	9	String	Yes

Index	Column Name	Description	Max Length	Type	Required
70	Service	Service description.	12	String	Yes
71	XRefStartDate	Date when the relationship began. Format YYYY-MM-DD. If this value is not provided it will be assumed to be the date the record is received.	10	Date	
72	XRefEndDate	Date when the relationship ended. Format YYYY-MM-DD. If this value is not provided, it will be assumed to be ongoing.	10	Date	

4 Appendixes

4.1 Assumptions

Service Elements and specific implementation values to be determined and documented during the Sandata Business Rules Process.

4.2 Other Important Points to Note

This interface is intended to be used in conjunction with the other Sandata Standard Open EVV interfaces. Exact interfaces and values to be used will be determined during implementation and appropriate companion guides will be developed with Payer / Program specific rules and values.

In the event of any required changes to the functionality covered in this document or the functionality already present in the code, it is recommended that a formal change control process be followed so as to ensure a set process for planning and scheduling, implementation of the same, verification and validation and roll-out for user testing.

4.3 Legend

Legend	
Field Name	Other Possible Naming
Client	Individual Member Patient Recipient
Employee	Caregiver Consumer Directed Worker Home Health Aide Staff Worker
Provider	Agency Third Party Admin (TPA)

Legend	
Field Name	Other Possible Naming
Payer	Admission Contract Insurance Company Managed Care Organization (MCO) State
Contract	Program Program Code
HCPCS	Bill Code Procedure Code Service

4.4 Acronyms and Definitions

Term	Definition
AKA	Also Known As
API	Application Programming Interface
GMT	Greenwich Mean Time
HTTP	Hyper Text Transfer Protocol
TBD	To Be Determined
UTC	Universal Time Coordinated

4.5 Time Zone List

This is the common list of time zone we used. If your area is not covered by this list please contact Sandata support to get additional time zone value that we accept. Please note that the value sent must exactly match the value and case shown.

Text Value	Daylight Saving
US/Alaska	Active

Text Value	Daylight Saving
US/Aleutian	Active
US/Arizona	Inactive
US/Central	Active
US/East-Indiana	Active
US/Eastern	Active
US/Hawaii	Inactive
US/Indiana-Starke	Active
US/Michigan	Active
US/Mountain	Active
US/Pacific	Active
US/Samoa	Inactive
America/Indiana/Indianapolis	Active
America/Indiana/Knox	Active
America/Indiana/Marengo	Active
America/Indiana/Petersburg	Active
America/Indiana/Vevay	Active
America/Indiana/Vincennes	Active
America/Puerto_Rico	Inactive

5 References

5.1 DSV

- [Wikipedia - Delimiter-separated values](#)

5.2 End of Line (EOL)

- [Wikipedia - Newline](#)

5.3 Error Detection

- [Wikipedia - SHA-1 Hash Function](#)
- [Wikipedia - Error Detection and Correction](#)

5.4 UTF-8

- [Wikipedia - UTF-8](#)

5.5 OpenPGP

- [Wikipedia - Pretty Good Privacy](#)
- [OpenPGP Website](#)
- [RFC4880](#)