



---

# Requirement Specification for Authorizations (AUTH) Part of the Open EVV Series of Interfaces

Version 7.2

**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](mailto:info@sandata.com)

Web: [www.sandata.com](http://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 Authorization (Auth) Interface ..... 7**
- 3 Data Exchange ..... 7**
  - 3.1 *Real-Time Transactions* ..... 7
  - 3.2 *Delimiter Separated Values (DSV)* ..... 12
- 4 Appendixes ..... 24**
  - 4.1 *Assumptions* ..... 24
  - 4.2 *Other Important Points to Note* ..... 24
  - 4.3 *Legend* ..... 24
  - 4.4 *Acronyms and Definitions* ..... 25
  - 4.5 *Time Zone List* ..... 26

## Revision History

Version	Description	Date Updated
7.1	Updated max length of "AuthorizationMaximum" field. Marked "ClientIdentifier" as required. Corrected value to be sent for "Suspended" status authorizations. Removed extra "DiagnosisCode" object in JSON example. Removed references to consuming "Full" files – leaving incremental as only option.	5/20/2019
7.2	Moved fields into the authorization limits segment where appropriate (service, modifiers, and program related fields)	6/12/2019

# 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 Authorization 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, required fields and those fields which will be omitted or left to the sender’s discretion.

## 1.1 Intended Audience

The intended audiences of this document are:

- 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 <b>date</b> and <b>time</b> 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
Timezone	<p>All time for tracking visits will be in UTC.</p> <p><b>Timezone</b> values are based on 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</p>	<p>A complete list of time zones can be found at: <a href="https://www.iana.org/time-zones">https://www.iana.org/time-zones</a></p> <p>See Appendix for list of EVV supported Timezones</p>

Data Type	Description	Example
	<p>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>	
String	<p>A <b>string</b> 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 <a href="#">Wikipedia String</a>)</p>
Integer	<p>An <b>integer</b> is a numeric value without a decimal. Integers are whole numbers and can be positive or negative.</p>	<p>52110 (positive) -87721 (negative) (See <a href="#">Wikipedia Integer</a>)</p>
Decimal	<p>A floating point number is referred to as a <b>decimal</b>. Can be positive or negative.</p>	<p>8221.231 (positive) -71.214 (negative) (See <a href="#">Wikipedia Decimal</a>)</p>
Boolean	<p>A logic predicate indicator that can be either true or false.</p>	<p>True False See <a href="#">Wikipedia Boolean</a></p>

## 2 Authorization (Auth) Interface

This specification is intended for a payer or provider to provide authorization information to Sandata regarding which clients are receiving specified services, the amount and the period during which the services are to be provided. This may also be referred to as Prior Authorization or Service Plan.

## 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

Authorizations may be sent via a real-time RESTful API for processing. Sandata will take each request as it is received, process the authorization 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.

Customers 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 customer 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 authorizations, 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 customer systems.

### 3.1.3 Authorizations Real-Time Processing

The authorizations real-time processing interfaces refers to Sandata's RESTful HTTPS endpoints for receiving authorizations. Sandata will provide the customer with URL endpoints for UAT and Production. Sandata will also provide the customer with a username and password that will use Basic Authentication to validate the request. The customer 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 3.2.18 for a detailed description of each field.

```
[
  {
    "PayerID": "Zitest",
    "PayerRegion": "NA",
    "ClientQualifier": "ClientSSN",
    "ClientIdentifier": "123456789",
    "ProviderQualifier": "SandataID",
    "ProviderID": "999999999",
    "AuthorizationReferenceNumber": "8725",
    "AuthorizationAmountType": "H",
    "AuthorizationMaximum": 123,
    "AuthorizationStartDate": "2018-09-10",
    "AuthorizationEndDate": "2018-09-22",
    "AuthorizationShared": "N",
```



```
"AuthorizationComments": "Sample comments",
"AuthorizationLimitType": "W",
"AuthorizationStatus": "A",
"AuthorizationLimit": [{
    "AuthorizationServiceID": "1015",
    "PayerProgram": "Testabc",
    "AuthorizationBillingType": "Test101",
    "AuthorizationLimit": "123",
    "AuthorizationWeekStart": "Sun",
    "AuthorizationLimitDayOfWeek": null,
    "AuthorizationLimitStartTime": "0600",
    "AuthorizationLimitEndTime": "1100",
    "Modifier1": "G1",
    "Modifier2": null,
    "Modifier3": null,
    "Modifier4": null
}],
"DiagnosisCode": [{
    "ClientDiagnosisCodeIsPrimary": "Y",
    "ClientDiagnosisCode": "12345",
    "ClientDiagnosisCodeBeginDate": "2018-01-01",
    "ClientDiagnosisCodeEndDate": "2018-01-01"
}]
}
]
```

## 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 HttpResponseMessage 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",  
  
    "ClientIdentifier": "12345",  
  
    "ClientQualifier": "12345",  
  
    "PayerID": "12345",  
  
    "PayerProgram": "12345",  
  
    "AuthorizationServiceID": "12345",  
  
    "AuthorizationReferenceNumber": "12345",  
  
    "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)**

Formats that use DSV to store two-dimensional arrays of data by separating the values in each row with specific delimiter characters. Most database and spreadsheet programs are able to read or save data in a delimited format. Due to their wide support, DSV files can be used in data exchange among many applications.

A delimited text file is a text file used to store data, in which each line represents a single record (i.e. Authorization) 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 delimited file 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 must be enclosed with double quotes ("").

Example: "<PAYER>"|"HHS"|"North America"|"MedicaidID"|"123456789"

### 3.2.4 Character Encoding

Each field within the delimited 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 DSV 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 Field Names)
- Customers, at their discretion, may exclude non-required fields.

Example: "PayerID"|"PayerProgram"|"PayerRegion" | ... | "ClientDiagnosisCodeEndDate"

### 3.2.6 File Naming Convention

The file naming convention was agreed upon during implementation, and is important to help with validation, entity mapping, dates and times to make sure files are not overwritten and 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 ("<PAYER>\_EVV")

- [EntityName] is the name of the domain specific name of the parent entity that reflects the data fields within the DSV file (“PriorAuth”)
- [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
  - [.pgp] signifies that the file has been encrypted with PGP [See [File Encryption](#)]
- Example format
  - <PAYER>\_EVV\_PriorAuth\_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 File

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)
      - StartDateime: The start date and military time when the records in the DSV were queried from. (See Date Time Format) [Optional]
      - EndDateime: 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)
  - StartDateime: The start date and military time when the records in the DSV were queried from. (See [Date Time Format](#))
  - EndDateime: 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 Payer outbound control file:

<PAYER>\_EVV\_Outbound\_ControlFile\_20180817.dsv.gpg

"FileName"|"RecordCount"

"<PAYER>\_EVV\_Provider\_20181002.dsv"|"2012"

"<PAYER>\_EVV\_Member\_20181002.dsv"|"12"

"<PAYER>\_EVV\_PriorAuth\_20181002.dsv"|"22"

"<PAYER>\_EVV\_Outbound\_ControlFile\_20181002.dsv"|"5"

"1/1/2010 3:19:01 PM" - "10/2/2018 3:47:43 PM"

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 for possible future use in regeneration efforts. Not expected to be validated by Sandata.

- Example inbound control file:
    - <PAYER>\_EVV\_Inbound\_ControlFile\_20180817.dsv.gpg
- "FileName"|"RecordCount"|"StartDateTime"|"EndDateTime"|"Hash"|"Success Count"|"Failed Count"
- "<PAYER>\_EVV\_Auth\_Errors\_20180817.dsv.gpg"|"2012"|"2018-09-18T00:00:00Z"|"2018-09-18T23:59:59Z"|"cjqqr032alimp883jasddejkm"|"2012"|"0"

### 3.2.10 Error File

#### PAYER AND SANDATA ERROR HANDLING PROCESS

- Sandata will notify PAYER 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 recipient email address provided by the Payer.
  - E.g. EVV\_Interface\_<payer>@<payer\_domain>.com
- The email Subject would include "<PAYER>-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
  - <PAYER>\_EVV\_Auth\_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 the Payer. 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 Payer’s secure SFTP, in a folder created specifically for Sandata, “/Sandata/<Payer\_Folder\_Name>”.

### **3.2.13 File Frequency**

Sandata will accept files on a daily schedule. The ongoing daily job will run and deliver files at about 2AM, Monday through Friday. For an authorization to be applied to a provider, the client information must be received at least once. If there is no matching client, the authorization will be held until the client is received.

### **3.2.14 Authorization File Logic**

Sandata is expecting to receive only active authorizations for clients who are part of a program and for services considered in scope for this project. Once an authorization is transmitted to Sandata, the authorization should only be resent if there are changes to be applied to the prior information.

### **3.2.15 Authorization Matching**

Unless otherwise specified during implementation, received authorizations will be matched against existing authorizations based on the following values:

- PayerID;
- PayerProgram if applicable for the implementation;
- PayerRegion if applicable for the implementation;
- ClientIdentifier;
- AuthorizationServiceID;
- AuthorizationReferenceNumber; and
- AuthorizationStartDate.

If the import process finds a matching authorization based on the above criteria, the authorization will be updated. If there is no match, the authorization will be added.

### **3.2.16 Shared Authorization**



Based on program requirements, Sandata can support the concept of a shared or umbrella authorization where one set of units or dollars is applied for a client across a set of services. It then is up to the discretion of the agency provider regarding which services to provide. The interface provides for receiving the authorization for each service that is shared.

- A shared authorization will consist of all authorization records that have the shared indicator (AuthorizationShared) set to 'Y', and the same
  - PayerID;
  - ClientIdentifier;
  - AuthorizationReferenceNumber; and
  - AuthorizationStartDate.
- For Shared Authorizations:
  - If a matching PayerID, ClientIdentifier, AuthorizationReferenceNumber and AuthorizationStartDate is found, the existing authorization will be updated.
  - If the following set of fields cannot be found in combination: PayerID, ClientIdentifier, and AuthorizationReferenceNumber a new authorization will be created.
  - If the following set of fields is found in combination: PayerID, ClientIdentifier and AuthorizationReferenceNumber but the associated AuthorizationStartDate is different, the incoming authorization will be rejected.
  - Limitations on shared authorizations must be the same for all services making up the shared authorization (daily, weekly, monthly, yearly or none).
  - Consumption of the authorization will be measured based on the unit of measure defined on the authorization (hourly, unit, visit, or monetary).

Utilization of both unit based and monetary based limits will be based on the amount billed by the provider.

### **3.2.17 Authorization 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.

### 3.2.18 Parent-Child Relationships

- The authorization 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:
  - AuthorizationLimit
  - DiagnosisCode
- If the DSV row represents either of the above segments, it is expected that the “SegmentName” is populated and the only data in the row is the basic authorization data and the specified segment. Note that, for authorizations with a single limit record, the data can be provided in a single row with SegmentName = AuthorizationLimit.
- If segments beyond basic authorization 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
1	PayerID	Sandata EVV assigned ID for the payer. Payer ID is determined during the implementation process.	64	String	Yes
2	PayerRegion	If applicable, the region in which this client is being provided services. Potential use and list of values to be determined during implementation.	2	String	
4	ClientQualifier	Value being sent to unique identify the client. Values: ClientSSN; ClientOtherID, ClientCustomID.	30	String	Yes
5	ClientIdentifier	ID used to uniquely identify the client. ID type identified by ClientQualifier.	64	String	Yes
6	ProviderQualifier	Identifier being sent as the unique identifier for the provider. Values: SandataID, NPI, API, MedicaidID, TaxID, Taxonomy, Legacy, Other.	30	String	Yes
7	ProviderID	ID to uniquely identify the provider. ID type identified by ProviderQualifier.	64	String	Yes

Index	Column Name	Description	Max Length	Type	Required
8	AuthorizationReferenceNumber	Unique identifier for the authorization generated by the payer's source system.	30	String	Yes
9	AuthorizationAmountType	The type of authorized amounts being supplied. Values: H = Hourly, V = Visit, U = Unit, M = Monetary.	1	String	Yes
10	AuthorizationMaximum	The maximum number of the specified type to allow. If the authorization is active, 0 in this field indicates unlimited. This is the overall limit and may be further limited by the Authorization Limits fields below. This value was previously a "numeric" type.	6	Integer	
11	AuthorizationStartDate	Start date for the authorization.	10	Date	Yes
12	AuthorizationEndDate	End Date for the authorization.	10	Date	
13	AuthorizationShared	To be used, shared authorizations must be part of the program specifications. Denotes that the authorization being provided is part of a shared authorization set. Values: Y, N. Default = N.	1	String	
14	AuthorizationComments	Free text comments. Used by the payer to include additional information. If value provided is greater than the maximum, the text will be truncated to the maximum size allowed.	256	String	
15	AuthorizationLimitType	If the authorization has sub limits, the type of sub limits being detailed. Values: N = None, S = Specified Schedule, D = Day, W = Week, M = Month, Y = Year. Default = N.	1	String	Yes
16	AuthorizationStatus	The status of the authorization. Values: A = Active, I = Inactive, V = Voided, S = Suspended. Default = A.	1	String	Yes
17	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: AuthorizationLimit** : Sub Limits within the Authorization. If AuthorizationLimitType = N this section will be ignored. Allows specification of monthly sub limits as well as limits based on day of week, time of day or specific schedules.

Index	Column Name	Description	Max Length	Type	Required
18	AuthorizationServiceID	The procedure or service to be performed. This will correspond to a HCPCS code to be used in billing. These values will be established as part of the implementation process.	5	String	Yes
19	PayerProgram	If applicable, the program to which this client belongs. Potential use and list of values to be determined during implementation.	9	String	Yes
20	AuthorizationBillingType	Optional information which may control additional mapping including determining billing information including HCPCS code and modifiers. Potential use and values to be determined during the implementation process. <u>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.</u>	64	String	
21	AuthorizationLimit	The limit value for the sublimit being specified. Required if this segment is provided. 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.	5	String	
22	AuthorizationWeekStart	Required if AuthorizationLimitType is W. Values: Mon, Tue, Wed, Thu, Fri, Sat, Sun.	3	String	
23	AuthorizationLimitDayOfWeek	Required if AuthorizationLimitType is S or D. Values: Mon, Tue, Wed, Thu, Fri, Sat, Sun.	3	String	
24	AuthorizationLimitStartTime	Required if AuthorizationLimitType is S. Optional if AuthorizationLimitType is D. If AuthorizationLimitType = S, the schedule MUST start at the time specified. If AuthorizationLimitType = D, the start and end times provide a range in which services are to be delivered. Does not apply for other AuthorizationLimitTypes. Format hhmm assuming a 24-hour clock. client or Account time zone is assumed.	4	String	

Index	Column Name	Description	Max Length	Type	Required
25	AuthorizationLimitEndTime	Optional if AuthorizationLimitType is S. Optional if AuthorizationLimitType is D. If AuthorizationLimitType = S, the service is scheduled to end at the time specified. If AuthorizationLimitType = D, the start and end times provide a range in which services are to be delivered. Does not apply for other AuthorizationLimitTypes. Format hh:mm assuming a 24 hour clock. client or Account time zone is assumed.	4	String	
26	Modifier1	First modifier if applicable. Authorizations may include modifier information to be used for service provision and billing.	2	String	
27	Modifier2	Second modifier if applicable. Authorizations may include modifier information to be used for service provision and billing.	2	String	
28	Modifier3	Third modifier if applicable. Authorizations may include modifier information to be used for service provision and billing.	2	String	
29	Modifier4	Fourth modifier if applicable. Authorizations may include modifier information to be used for service provision and billing.	2	String	
<b>Segment – DiagnosisCode</b> : Diagnosis codes for the client. This is a repeating element. Only one diagnosis code can be indicated as primary.					
30	ClientDiagnosisCodelsPrimary	Indicates that the diagnosis code is the primary code for the client. If more than one code is primary, first value designated as primary will be assumed to be primary. If no value is provided as primary, first value provided will be assumed to be primary. Required if segment is provided. Values: Y, N. Default = N. 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.	1	String	

Index	Column Name	Description	Max Length	Type	Required
29	ClientDiagnosisCode	The client's diagnosis code in ICD-10 format. 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	
30	ClientDiagnosisCodeBeginDate	The beginning date associated with the specified diagnosis code. Format YYYY-MM-DD. 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	Date	
31	ClientDiagnosisCodeEndDate	The ending date associated with the specified diagnosis code. Format YYYY-MM-DD. 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	Date	

Additional elements such as ClientDevice, EmployeeKey, Service, etc. may be added for a particular program and can be found in their companion guide.

## 4 Appendixes

### 4.1 Assumptions

N/A

### 4.2 Other Important Points to Note

In the event of any required changes to the process apart from 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)
Payer	Admission Contract Insurance Company Managed Care Organization (MCO) State



Legend	
Field Name	Other Possible Naming
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	Hypertext Transfer Protocol
SRS	System Requirement Specifications
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
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