



Sandata Specification User Guide

Delimited Flat Files



July 30, 2021
Sandata Technologies, LLC
sandata.com

Clella Newcomb, Director Implementation Services
cnewcomb@sandata.com
516.484.4400 X4324

Table Of Contents

Sandata Specification User Guide	1
Delimited Flat Files	1
Revision History	4
Specification User Guide – Delimited Flat Files	5
Overview	5
Intended Audience	5
Transmission Frequency	5
Transmission Limits	5
Data Type Format Details.....	6
Delimiter Separated Values (DSV).....	6
Supported Delimiters	7
End of Line Characters	7
Double Quotes	7
Character Encoding.....	7
Header Record	7
File Naming Convention	8
File Encryption	9
Cryptographic Hash (Optional).....	9
Control Files	9
File Processing	11
Rejected File Handling	11
Rejected Record Handling	11
Error Handling for Outbound Files	12
File Retention	12
File Transport.....	12
File Location	13
Specification Reading Tips	13
Example Alt EVV SegmentName: Provider Identification.....	13

Parent-Child Relationships	13
Specification Specific File Logic	14
Member	14
Authorization	15
Authorization Matching	15
Shared Authorization	15
Terminology	16
Legend	16
Acronyms	17

Revision History

Date	Version #	Author	Description of the Document Updates	Requires Update to Business Rules?
7/30/2021	1.0	Clella Newcomb	Document Creation	Yes
3/22/2022	1.1	Jason Feder	Moved DWE references to a separate guide	No

Specification User Guide – Delimited Flat Files

Overview

This specification user guide is intended to document the requirements for using the Sandata Flat File Interface to provide program information to Sandata.

Program specifications are created for each customer implementation to specify agreed upon frequencies, expected values for fields and those fields which will be omitted or left to the sender's discretion.

This document contains technical information for data transmission.

Intended Audience

The intended audience of this document is:

- Project Management and Technical teams at the state, 3rd party vendors or agencies who will be implementing this interface.

Transmission Frequency

Inbound

Sandata will accept files at any time of the day 365 days a year. The ongoing processing job will run 1 time per hour where we sweep the designated sFTP for new files and if found will attempt to process the new file.

Outbound

When Sandata receives Inbound files on an sFTP we will attempt to process then will return response files back to the sFTP. The files will be stored in an agreed upon folder on the same sFTP. Depending on processing times and size of the file received, the response files will typically be returned to the sFTP within 1 hour.

Transmission Limits

A single file may contain from zero to a maximum of 50,000 records.

If the file size exceeds the maximum limit listed above, then the entire file may fail to process.

During peak loads, files received may be queued and processed as resources permit. Other files received for the specification will be queued and processed in order.

Data Type Format Details

On fields where the specification denotes that one of a specific set of reference values is to be provided, the value sent must exactly match the value and case shown.

Data Type	Description	Example
DateTime	The date and time is represented as a string with the following format: YYYY-MM-DDTHH:MM:SSZ All times will be provided in UTC. If time is not material, provide as expected.	2016-12-20T16:10:28Z
Date (only Date)	The data is represented as a string with the following format: YYYY-MM-DD Date only will be sent in UTC format.	2016-12-20
Timezone	All time for tracking visits will be in UTC. The Time zone name expected in each transaction is the actual Time zone where the event took place. i.e. US/Eastern	A complete list of time zones can be found in the appendix of this document.
String	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).	"This is a string" (See Wikipedia String)
Integer	An integer is a numeric value without a decimal. Integers are whole numbers and can be positive or negative.	52110 (positive) -87721 (negative) (See Wikipedia Integer)
Decimal	A floating point number is referred to as a decimal. Can be positive or negative.	8221.231 (positive) -71.214 (negative) (See Wikipedia Decimal)
Boolean	A logic predicate indicator that can be either true or false.	True False See Wikipedia Boolean

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 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 file 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](#))

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

End of Line Characters

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

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”

Character Encoding

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

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, including casing. (See DSV Field Names)
- Customers, at their discretion, may exclude non-required fields.

Example: "ProviderID"|"ProviderQualifier"|"ProviderName"|"PayerID"

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.

- [EntityName]_[Prefix]_[YYYYMMDD]_[HHMMSS.SSS]_[FileExtensions]
 - [EntityName] is the name of the domain specific name of the parent entity that reflects the data fields within the DSV file ("Member" | "Provider")
 - [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")
 - [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
- [.pgp/.pgp] signifies that the file has been encrypted with PGP [See [File Encryption](#)]

Example Format: <SENDER>_EVV_Member_20180817_070156.006.dsv.zip.gpg

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 “pgp” file extension.

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](#))

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_20181002_070152.223.dsv.gpg
"FileName"|"RecordCount"
"SENDER_EVV_Provider_20181002_070152.223.dsv"|"2012"
"SENDER_EVV_Outbound_ControlFile_20181002_070152.223.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_20181002_070152.223.dsv.gpg
"FileName"|"RecordCount"|"StartDateTime"|"EndDateTime"|"Hash"|"Success Count"|"Failed Count"
"<SENDER>_EVV_Provider_Errors_20181002_070152.223.dsv.gpg"|"2012"|"2018-09-18T00:00:00Z"|"2018-09-18T23:59:59Z"|"cjpqr032alimp883jasddejkm"|"2012"|"0"
```

File Processing

Rejected File Handling

Sandata will process the entire file. If the file does not pass validations, it will be rejected in its entirety. It is expected that a designated representative will be identified who will be contacted if any issues are found and an escalation process will be developed. It is expected that a corrected file will be resubmitted on the same day with a new file name.

Rejected Record Handling

If any of the required fields are not included in a given record, within the incoming file, the record will be rejected. When a record gets rejected, a record log of the instance will be created by Sandata and saved to a specified (SFTP or otherwise agreed upon) location.

Error Handling for Outbound Files

- Sandata can notify Sender via email to alert of any errors found in processing each file that was imported. This is not usually turned on as the Error file will be automatically sent to the sFTP indicating the errors.
- Sandata will not send emails or error files if there are no errors detected for the delivery.
- Error files will not be sent in the case where the control file indicates that 0 records have been sent in the outbound file.
- The email would be addressed to <Spec type> SENDER@SENDER_DOMAIN.com
- Sandata will provide the customer with an error file for each file that was imported only if at least 1 error occurred within the file.
- 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
Multiple errors may be added to this field when multiple errors are present.
- 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.zip.gpg

File Retention

Sandata will retain the file for a period of no less than 10 days or as stated in the Business Rules should an issue be detected with the file copied to the agreed upon SFTP site. Each customer can request the file be manually recopied if needed. The file naming convention has been 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.

File Transport

Files will be consumed and delivered via Secure FTP (SFTP). The target sFTP server will be hosted by Sender. The hostname, username, password or SSH Keys are discussed and tested during implementation.

File Location

DSV files will be located on the sFTP server, in a folder created specifically for each file type. Example: “/Prod/From_SENDER”

Specification Reading Tips

- Segments are denoted as required or optional in the applicable specification and are a named field for transmission.
Example Alt EVV SegmentName: Provider Identification
- The fields listed in the specification are the fields available for transmission to Sandata EVV.
- All data elements in the specification will indicate if NULL is an acceptable value in the Validation rules column. If the element does not state that NULL is an acceptable value then that element is required. In .dsv files the NULL value should be represented as |””|
- Required elements 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. This means that the element could be completely omitted from the header and the data rows if desired.
- As part of the implementation process, required fields may be adjusted and the available fields may be reduced based on the program specifics.
- All element formatting is case sensitive. All field names must be provided in EXACTLY the casing used in the definitions below. Specifically, this includes reference values denoted in “double quotes”.

Parent-Child Relationships

- The data represented in the DSV file naturally has some parent-child (one-to-many) relationships
- These can be referenced in the flat file using the “SegmentName” field
Example Alt EVV SegmentName: Provider Identification

- If the DSV row represents basic 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 data and the specified segment
- If segments beyond basic data are provided, there are specific fields that must be included for each. See the specification for required fields per segment. Note the segment descriptions will 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.
- In the case where the “SegmentName” field is in use to denote child data rows, it is expected that these rows are contiguous and immediately follow the parent row within the file. In addition, when a parent record is being sent with data for multiple child segments (e.g. ClientAddress, ClientPhone, etc...), the records for each child segment must also be contiguous. For example, all ClientAddress rows follow the parent, and all ClientPhone rows follow the ClientAddress rows, and so on. The order of the child segments is immaterial, so long as the data for each segment remains contiguous.

Specification Specific File Logic

Member

- 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 present if there are changes to be applied to the prior information.
- Valid Member segment names are:
 - ClientDesignee
 - ClientEligibility
 - ClientContact
 - ClientAddress

- ClientPhone
- ClientWorkerXref

Authorization

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.

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.

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.

Terminology

Legend

Field Name	Other Possible Naming
Client	Individual
	Member
	Patient
	Recipient
Employee	Caregiver
	Consumer Directed Worker
	Home Health Aide
	Staff
	Worker
Provider	Agency

Payer	Third Party Admin (TPA)
	Admission
	Contract
	Insurance Company
	Managed Care Organization (MCO)
Contract	State
	Program
HCPCS	Program Code
	Bill Code
	<i>Procedure Code</i>
	<i>Service</i>

Acronyms

Term	Definition
AKA	Also Known As
API	Application Programming Interface
GMT	Greenwich Mean Time
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
SOAP	Simple Object Access Protocol
SRS	System Requirement Specifications
TBD	To Be Determined
UTC	Universal Time Coordinated
XML	Extensible Markup Language