



3rd Party Data Interface Specifications:

Standard Scheduling Interface

Electronic Visit Verification System



Version 3.1



July 29, 2016

Prepared by:

Judy Ross

Technical Lead

516.484.4400 x 1256

jross@sandata.com



Table of Contents

Overview	3
1. 3 rd Party Scheduler -> Sandata: Schedules.....	4



Overview

This document is intended to provide provider agencies (and their respective vendors if applicable) specifications for interfacing with the Sandata EVV system. This specification assumes that members and authorizations are already being received from a payer system and the functionality implemented for the payer includes the ability to support 3rd party interfaces.

Scheduling systems have the ability to provide to Sandata employee information and schedules. Scheduling may be disallowed for some programs if the schedules cannot be properly validated against the program rules.



1. 3rd Party Scheduler -> Sandata: Schedules

General Information

Source	Destination(s)	Type	Interface Name	Description	File Type	Timing
Various	SPM	REST web interface	Schedules	Allow Provider Agencies to supply their schedules to the system directly from their own scheduling system avoid the need for duplicate entry.	XML or JSON	At Agency Discretion

Any customers utilizing SAM as well as any other supplemental software for the purpose of day to day operations in the homecare market often find that duplicate entry for schedule information is necessary, and can often lead to incorrect or un-matching data between the two systems. This document can be generically utilized for agencies where the schedule import process has to be implemented. A standard import of these records from one system to the other will offer more data integrity and avoid additional data entry or modification for our system users.

File Format

The REST interface will accept data in JSON format that conforms to the OpenEVV Schedule specifications.

File Naming

N/A.

Frequency

Sandata will provide a REST web interface. The customer can send data as frequently as they need to the interface.

File Logic

All schedules provided will be considered valid and will not be checked against the authorization. However, if schedules are created and serviced outside of an existing authorization, they will not be billable.

- All required fields must be included and pass validation.
- A Client is required to import a Schedule.
- The listed Client and Staff must exist prior to importing the Schedule.
- The listed Client and Staff must be active for the date of the Schedule.
- The listed Client must have an Admission active for the date of the Schedule.
- The Admission type must be specified for a Client with multiple active Admissions.



- The specified service must match the service allowed by the Client’s Admission.
- The listed Staff must be allowed to perform the level of service specified.

Data Transition Plan

All employees should be sent in the initial feed to the system. Subsequently, only new employees and employees with changes since the last transmission should be sent.

Available Fields

The fields listed below are the fields available for import into Santrax Agency Management. The required columns are marked in orange, and the column must be present in the file or the import will be incomplete.

Required Columns

The required columns are marked in red, and the column must be present in the file or the import will abort.

The following are the fields that can be sent.

Fields	Datatypes	Descriptions
account	varchar (25)	SAM Customer Number. Required (Provider of the data)
Id	varchar (50)	User defined ID. Commonly used to store Schedule ID from another system. This value is used to match the Schedule to an existing record during import. Required
startTime	datetime	Start date and time of the schedule. (YYYY-MM-DD HH:mm:ss with dates in range 1753-01-01 through 9999-12-31 and time in range 00:00:00 through 23:59:59) Required
endTime	datetime	End date and time of the schedule. (YYYY-MM-DD HH:mm:ss with dates in range 1753-01-01 through 9999-12-31 and time in range 00:00:00 through 23:59:59) Required
clientId	varchar(25)	User defined ID of Client for the Schedule. This value is used to match the Client of an existing record. This must match value of Client Other ID in SAM. Required .
employeeId	varchar(25)	User defined ID of Staff for the Schedule. This value is used to match the Staff of an existing record. This must match value of Staff agency ID in SAM. Required .
service	varchar(6)	Identifier for service performed. This ID must already be set

Fields	Datatypes	Descriptions
		up in SAM application. .(required)
admissionType	varchar(3)	This must be setup in application prior to import. (required if client has multiple admission types)
company	varchar(6)	User-Defined code (Companies -> ID). This must match Company ID value as defined in SAM. .(required)
Status	varchar(2)	Identifier for the status of the Schedule. (01 = Created/Updated, 10 = Cancelled).
cancellationReasonCode	varchar(6)	Identifier for cancellation reason as defined in SAM. .(required for cancel status)
comments	varchar(1000)	Comments on Schedule.
eventCode	Varchar(6)	Event code for Schedule as defined in SAM. If no event code is provided, the agency's default event code will be used.
clientScheduleTypeId	varchar(2)	Identifier for Schedule type. (01 = Hourly, 02 = visits, 05 = units) Required
employeeScheduleTypeId	varchar(2)	Identifier for Schedule type. (01 = Hourly, 02 = visits, 05 = units) Required if employee is on schedule.

Sample Data Templates

Schedule objects should be sent in JSON format. Additional examples can be provided upon request.

A.1 JSON

```
{
  "account": "1234",
  "id": "123456789",
  "startTime": "2014-03-14 09:00",
  "endTime": "2014-03-14 10:00",
  "clientId": "0123452",
  "employeeId": "52345642",
  "service": "SN",
  "admissionType": "ADW",
  "company": "9999",
  "status": "01",
  "cancellationReasonCode": null,
  "comments": null,
  "eventCode": "01",
  "clientScheduleTypeId": "02",
  "employeeScheduleTypeId": "01"
}
```



*Schedule.cancellation and Schedule.comments were removed for this example.

Response

- The import result will be returned in the data of the Response object.
- It will contain a text summary of the result of the import.
- The number of records successfully imported.
- The number of records that encountered errors and were unsuccessfully imported.
- A key-value map containing the ID of the record and a detailed message of the error or errors encountered.

A.1 JSON

```
{
  status: "SUCCESS"
  token: null
  messageSummary: null
  messageDetail: null
  errorMessage: null
  data:
    {
      importResultSummary: "0 out of 1 schedules successfully queued."
      successfulRecordCount: 0
      erroneousRecordCount: 1
      erroneousRecords:
        {
          0:
            {
              recordId: "123456789"
              message: "ERROR: Service is required. "
            }
        }
    }
}
```