Document #: 12021

Status: DRAFT

Version 1.2

Approval date: 08-20-2012

Updated: 04-20-2015

SunSpec Model Data Exchange

SunSpec Alliance Interoperability Specification

Contributors:

Lynn Linse, Bob Fox, John Blair, Howard Davis, Aaron McNally, Stephen Lapointe, Richard Duong, Robert Hyatt, Bill Randle, Martin Beran, Robert Orozco, Michael Murphy, Eric Valleton, Justin Robinson, John Nunneley, Tom Tansy

Abstract

The SunSpec Alliance Interoperability Specification suite consists of the following documents:

- SunSpec Technology Overview
- SunSpec Information Model Specifications & Reference
- SunSpec Data Exchange Specifications

This document describes the format for exchanging SunSpec Information Model data using XML. This exchange typically occurs between a Logger and a monitoring Host. A RESTful HTTP interface for transporting the data is also defined.

License Agreement and Copyright Notice

Prepared by the SunSpec Alliance
4030 Moorpark Avenue, Suite 109
San Jose, CA 95117

Website: www.sunspec.org
Email: info@sunspec.org

Revision History

Revision	Date	Reason
1.0		Proposed final draft for release
1.0a		Repair some broken reference links – no substantive change to text.
1.0b	08-20-2012	Update examples to use V2 model point names (snam). Note that V1 names must still be supported. Use D_FAILURE instead of DR_FAILURE as per sunspec_ws XSD. Update examples to be compliant with XSD. Remove XSD from this document – it is distributed as a separate file. errorDetail element has been removed from the sunspec_ws XSD as per the examples in this document, e.g <code />, <reason />, and <message /> are peers of <status /> and direct sub elements of dResult.
1.1	09-18-2014	Change title Reformat to match SunSpec new document template Update terminology
1.2	04-21-2015	Removed logger configuration Corrected examples Edited for readability

About the SunSpec Specification Process

SunSpec Alliance specifications are initiated by SunSpec members desiring to establish an industry standard for mutual benefit. Any SunSpec member can propose a technical work item. Given sufficient interest and time to participate, and barring any significant objections, a workgroup is formed and its charter is approved by the board of directors. The workgroup meets regularly to advance the agenda of the team.

The output of the workgroup is generally in the form of an Interoperability Specification. These documents are considered to be normative, meaning that there is a matter of conformance required to support interoperability. The revision and associated process of managing these documents is tightly controlled. Other documents are informative, or make some recommendation with regard to best practices, but are not a matter of conformance. Informative documents can be revised more freely and frequently to improve the quality and quantity of information provided.

SunSpec Interoperability Specifications follow this lifecycle pattern of DRAFT, TEST, APPROVED and SUPERSEDED.

For more information or to download a SunSpec Alliance specification, go to http://www.sunspec.org/specifications.

Nomenclature

Abbreviation	Meaning
DID	Device Identification
DNS	Domain Name System
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
MAC	Media Access Control
RFC	Request For Comment
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
Md	SunSpec Common Model: manufacturer's model number
Mn	SunSpec Common Model: manufacturer's name
NAT	Network Address Translation
OAuth	Open standard to Authorization (see www.oauth.net)
SF	Scale Factor
SN	SunSpec Common Model: manufacturer's serial number
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
X.509	ITU-T standard for a Public Key Infrastructure and Privilege Management Infrastructure
XML	Extensible Markup Language

1. INTRODUCTION

The SunSpec Alliance Interoperability Specifications describe the format of data transported to diverse third party host tools. This Data Model Exchange document defines how SunSpec data models (such as meter or inverter models) are represented in XML for exchange

It is most typically used for conveying model data from plant-resident Loggers to central historian Hosts, but can also be used to extract model data from monitoring systems.

2. TERMS AND CONCEPTS

2.1 Device

The **Device** is a local asset - such as a meter, inverter or environmental device – whose monitored data points are collected periodically by a Logger and transmitted to a Host for long term storage and processing. In SunSpec terms, a **Device** is a physical entity, whose monitored data points are composed of one or more logical SunSpec device blocks. A typical device shall contain a SunSpec common block, and one or more SunSpec device blocks. The monitored data points are defined within the device blocks. See the SunSpec Information Model Specifications for further details on **Devices**. **Logger** vendors are free to manage non-compliant local devices as long as the data uploaded to the **Host** fully complies with this specification.

2.2 Device Identification

Each **Device** must have a globally unique identifier. Two methods are supported by SunSpec.

Common Block Data

The preferred method is to include the three Common Block fields (C_Manufacturer, C_Model, and C_SerialNumber) as the device identification. Full compliance with the SunSpec specification requires the combination of these 3 data items to be globally unique, giving the host the ability to look up considerable knowledge by other means.

For example, the device would be represented as the three attributes: **man="gsc" mod="r300" sn="123456"**. See section 3.4 for attribute descriptions.

Logger-specific, user assigned id

The alternative method is to support direct entry of a device identification string within the logger. This id must be unique within the context of a single logger, so when combined with the logger id it becomes unique system-wide. An optional interface attribute maybe enabled in the logger, which allows the logger to specify a device by interface (as if="X") and device (as id="X").

For example, the device would be represented as the attributes **lid="11:22:33:44:55:66" id="A1234"**. See section 3.4 for attribute descriptions.

2.3 Host

The **Host** is the remote application system which expects data to be uploaded from **Loggers**. It can also be called a 'server' or 'monitoring system'. More than one **Host** can exist for a specific **Logger**.

2.4 Logger

The **Logger** collects data from one or more **Devices**, storing it for periodic upload to remote **Hosts**. The means by which the **Logger** collects and locally stores data is outside the scope of this specification. This specification only covers how the data is encoded for the upload to the **Host**(s).

This version of the specification assumes the mechanism to configure the **Logger** is vendor specific, while specifying settings which the **Logger** is expected to support.

2.5 Logger Id

The **Logger Id** is a globally unique string which defines a single **Logger**.

2.6 Raw Data

Raw Data is collected and stored by the **Logger**. The format and meaning is vendor and implementation specific.

2.7 Encoded Data

Encoded Data is the data prepared by the Logger for upload – for example meter data encoded as XML. The format must comply with this specification. The encoded data is independent of the Transport selected, for example a **Logger** could save the **Encoded Data** as an XML file on a removable FLASH device, or use HTTPS or FTP to upload to a **Host**.

2.8 Transport

How the data is uploaded from Logger to Hosts is the process of **Transport**. The SunSpec Alliance shall be using standard industry protocols. This specification includes a transport definition using HTTP/POST and XML.

3. XML ENCODING OF DATA

3.1 XML Schema

The formal XML Schema is attached in the Appendix.

3.2 High Level Overview

A data upload from a Logger to the Host can be summarized as any number of Device records, each consisting of a collection of data point records. Here is a simplified XML example:

```
<SunSpecData v="1">
  <d ns="mac" lid="11:22:33:44:55:66" man="gsc" mod="r300" sn="123456"
t="2011-05-12T09:21:49Z" cid="2">
    <m id="101" x="1">
      <p id="A">30.43</p>
      <p id="PhVphA" sf="-1">2216</p>
      <p id="W" u="Watts">6701.3</p>
      <p id="Hz">60.01</p>
      <p id="WH">126973</p>
      <p id="DCA">14.28</p>
      <p id="DCV">469</p>
      <p id="DCW">6805</p>
      <p id="TmpOt">32.94</p>
      <p id="St">4</p>
    </m>
  </d>
</SunSpecData>
```

3.3 Element Tag Summary

Tag	M/O/C	Description
v	M	Version number attribute to SunSpecData block.
d	M	Device Record.
d.cid	O	Result Correlation, allowing a logger to request feedback from the host on a complete device record. All point data within this set must be accepted for this acknowledgement to be true.
d.id	O	Optional Device Id string, which overrides the d.man, d.mod, and d.sn values if present.
d.if	O	Optional Interface Id string, which only has meaning when the optional device id string (d.id).
d.lid	C	Logger Id string. Required by default, my be disabled by logger configuration if not required by Host.
d.man	M	Mn from the device's Common Block
d.mod	M	Md from the device's Common Block
d.ns	C	Doman Namespace for the Logger Id (d.lid)
d.sn	M	SN from the device's Common Block
d.t	M	Time-stamp in RFC 3339 format – see Standard Data Formats.
m	M	Model Record – defines the SunSpec Model used to understand the following group of data point records.
m.id	M	Model Id (referred to as DID or Device Model Identifier)
m.ns	O	Required for Vendor Extension Blocks and model blocks containing non-SunSpec data. Vendors are encouraged to use their official SunSpec Mn (C_Manufacturer) name for this.
m.x	O	Index used in aggregated devices.
p	M	Data Point Record
p.d	O	Description for the Data Point.
p.id	M	Point Name as defined in the appropriate SunSpec Device Model.
p.sf	O	Scale Factor to be applied to the point value.
p.t	O	Time-stamp in RFC 3339 format – see Standard Data Formats.
p.u	O	Unit of Measure, optional for Sunspec-aware host.
p.x	O	Index used in devices when variable like-named data points.

3.4 Clarification of element / attribute use

Device Record <d ...

Contains all model and data point records for a specific, identifiable device in the field assembled at one moment in time.

There may not be a one-to-one correspondence between Device Records and physical devices. For example, a solar inverter might be coupled with a collection of three simple environmental sensors on a proprietary bus. Depending on the Logger and software design, these could be considered 4 distinct devices, or a single device containing 4 distinct SunSpec Device Models.

Device Result Correlation Id <d cid="XXX" …

This optional attribute allows a host to acknowledge a successful or failed upload of a specific device record in the future. The string value in cid="xxx" is Logger-specific and should be unique enough within the operational memory of both the host and logger to prevent mis-operation due to a mistaken context.

A suggested format is a 5-digit number incrementing between 1 and 99999, however hosts should never attach any meaning to this string. The Host merely returns the value as a token, with the Logger understanding the context.

If the Logger does not include any such attributes, then host feedback elements cannot be correlated to specific device records that have identical identification attributes.

The method used by the Host to acknowledge the data is defined in section 6.

Device Identifier <d id="XXX" …

This optional attribute includes a logger-defined Device Identifier, which the logger vendor and/or the user configuring the logger need to insure are globally unique.

The Host should prefer the "id" attribute when available. However, the man/mod/sn attributes are required, so a host could use them exclusively – see 0 for more discussion on Device Identification.

Logger Identifier <d lid="XXX" …

This attribute is mandatory by default, but can be disabled within loggers when the host uses the Device Identifier exclusively.

The default Logger ID shall be an IEEE MAC address, which may be accompanied with a domain namespace attribute. For example ns="mac" lid="11:22:33:44:55:66".

IEEE MAC addresses shall be used for various identifying tasks. These may include 6-byte Ethernet, 8-byte 802.15.4/ZigBee forms, and other IEEE forms.

Values shall be formatted as a subset of standard IEEE-permitted forms. Encoding as text shall be limited to uppercase or lowercase, using the character ':' as a byte separator. For example, 1A:2B:3C:4D:5E:6F. This reduces the computational complexity of data importation, as well as reducing conformance and application test complexity.

Logger IDs other than MAC addresses must specify the domain namespace attribute – see the Domain Namespace Attribute in 0.

Device Common Model data: Manufacturer <d man="XXX" …

This mandatory attribute uploads the Mn value from the SunSpec Common Model.

This value is used as part of the Device Identifier, where a suggested database key would be to concatenate the three mandatory values Mn, Md, and SN. For example: attributes man="gsc" mod="r300" sn="123456" could produce the key "GSC-r300-123456".

Vendors creating SunSpec compliant devices (or translation/proxy devices) are responsible to insure the uniqueness of these values global to products worldwide.

Device Common Model data: Model <d mod="XXX" …

This mandatory attribute uploads the Md value from the SunSpec Common Model. This value is used as part of the Device Identifier.

Domain Namespace <d ns="XXX" …

This optional attribute defines the type of Logger Id encoded into attribute "lil". For Example: ns="mac" lid="11:22:33:44:55:66" or ns="ims" lid="12345678901234".

Tags	Description
dns	Static DNS name such as 'solar01.sunspec.org'. Users must insure the results are static and globally unique.
ims	Cellular IMS number such as used for unique module activation. This should only be used on a pure cellular device without an IEEE Ethernet MAC address.
ip	Non-Private, fixed public IP Address such as '67.23.0.1' or 'fe80::b8d5:4f44:f232:c27'. Users must insure the results are permanent and globally unique. Private and dynamic IP must not be used.
mac	IEEE MAC address as defined in Standard Data Formats
{vendor}	The unique C_Manufacturer value assigned by SunSpec for use within the Common Block may be used to implement vendor-specific globally unique ID strings. An example would be 'digi'.

Device Common Model data: Serial Number <d sn="XXX" …

This mandatory attribute uploads the SN value from the SunSpec Common Model. This value is used as part of the Device Identifier – see 0.

Timestamp <d t="XXX" …

This mandatory attribute defines when the logger assembled this <d> record.

Timestamps shall be formatted as a subset of IETF RFC 3339. All time shall be based upon UTC/Universal time and include the 'Z' marker. An example valid timestamp is "2011-04-11T16:02:50Z".

Optionally, a fractional seconds component can be included, but only if they are meaningful and not false-accuracy. For example: "2011-04-11T16:02:50.423Z". An example situation where

fractional seconds are NOT meaningful – the logger polls a multi-drop of serial slaves at 9600 baud. The time-stamp is applied by the logger based upon when the response clears various lower application and hardware layers. If jitter and latency are not tightly monitored and controlled by design, then claiming the data was valid at 0.423 (423 msec) after the second is false-accuracy since the data was really sampled at an unknown time before that moment.

Model Record <m …

This required sub-element below the <d> element defines the SunSpec Model used to understand the following group of data point records. SunSpec-aware hosts use the model id to 'look' up repetitive information such as "What is the unit of measure for data point A, from the SunSpec Device Model 103?"

Model Id <m id="XXX" …

This required attribute holds the Device Model Identifier (or DID) as defined in the various SunSpec Model specifications.

Note that vendors implementing Vendor Extension Blocks must register their DID with SunSpec to avoid conflict with other vendors.

Model Id Namespace <m ns="XXX" …

By default, the model id must be the values assigned by SunSpec, so the default namespace is "sunspec" (<m id="XXX" ns="sunspec" …).

Vendors who choose to upload non-SunSpec data under the SunSpec format may assign their own id as string, plus must include the namespace attribute to qualify the non-unique model id. Vendors are encouraged to use a NS attribute which is compatible with their Common Block Mn value, but that is not required.

Uploading non-SunSpec models must be a user option within the logger, so that users may disable or enable the upload as desired. A logger which does not allow disabling the upload of non-SunSpec model data shall be deemed non-conformant.

Model Record Index <m x="X" …

This optional attribute is used within aggregate devices, where a single physical device may include multiple instances of the same model.

Data Point Record <p …

This required sub-element below the <m> element defines one data point or sample.

Data Point Description <p d="XXX" …

This optional attribute adds a user-friendly description to a data point record for Vendor-Specific models. This attribute should NOT be included on standard SunSpec Device Models.

Data Point Id (Name) <p id="XXX" …

This required attribute is the name as defined in the appropriate SunSpec Device Model or Vendor extension.

Data Point Scale Factor <p sf="X" …

Some Sunspec model maps include scaled integers to represent 16-bit Modbus data by splitting the floating point value into a fixed value and power-of-tens multiplier. For Example: 234.5 watts may be represented as W = 2345 and W_SF = -1 (10^{-1} or a multiplier of 1/10th).

Generally, the Logger converts the data value and scale factor into a string representation of a floating point data value. Optionally, the logger can attach the "sf" attribute to force the host to do the calculation. If this attribute exists, then the data point text value is raw and unscaled. The "sf" attribute is to be assumed 0 if omitted, so 10^0 or a multiplier of 1.

Below are the two acceptable ways to encode scaled integers.

```
<p id="W">6701.3</p>
<p id="W" sf="-1">67013</p>
```

Note that Scale Factors are defined as separate data points in the SunSpec Model Definitions (e.g. W_SF), but are encoded as "sf" attributes to their corresponding data point in this encoding.

Data Point Timestamp <p t="X" …

This optional attribute allows the logger to assign a different timestamp to a data point than the timestamp in the device record. This timestamp shall be the time at which the data point was obtained.

It follows the same rules as the device record timestamp.

Data Point Unit of Measure <p u="XXX" …

This optional attribute allows a logger to assign a user-friendly unit-of-measure (such as "Watts" or "Volts"). By default, the unit of measure attribute is NOT included in the encoded data, and it is assumed the Host uses the model id and data point id/name to lookup the unit of measure.

A Sunspec-unaware Host may require the units attribute to be uploaded. Loggers are not required to support such hosts, but it is encouraged to include a logger configuration setting to enable their inclusion.

Hosts may ignore the units attribute, or treat it as informational only.

The Unit of Measure attribute may NOT be used to override the standard units in Sunspec models – for example, the inverter value "DCW" is defined as units "Watts". A logger must NOT send this data as <p id="DCW" u="KWatts"> and expect the host to do units conversion from kilo-watts to watts.

Data Point Record Index <p x="X" …

This optional attribute is used within devices, where a single physical device may include multiple instances of the same data point – for example a back-of-module temperature device which may have 4 inputs.

Clarification on SunSpec Common Model data

The 3 data values Mn, Md, and SN may be included within the Device Record as <d man="gsc" mod="r300" sn="123456" … >. If these values are used to form the device identification, then they must be included in every upload report cycle.

However, since the Common Model data is generally constant and unchanging, loggers may select not to upload all of the Common Model data every report cycle. SunSpec does not define

how loggers decide when to upload, however suggested examples could include once a day, when the value is detected to have changed, or anytime the logger restarts.

Common Model data is uploaded as any other SunSpec model, and the Mn, Md, and SN data should be included for consistency. Below is an example upload for a Generic-Solar-Corp inverter:

```
<SunSpecData v="1">
  <d man="gsc" mod="r300" sn="123456" t="2011-05-12T09:21:49Z">
    <m id="1">
      <p id="Mn">gsc</p>
      <p id="Md">r300</p>
      <p id="Opt">485-x4-sp</p>
      <p id="Vr">1.23.567</p>
      <p id="SN">123456</p>
      <p id="DA">1</p>
    </m>
    <m id="101">
      <p id="A">30.43</p>
      <p id="PhVphA" sf="-1">2216</p>
      <p id="W">6701.3</p>
      <p id="Hz">60.01</p>
      <p id="WH">126973</p>
      <p id="TmpOt">32.94</p>
      <p id="St">4</p>
    </m>
  </d>
</SunSpecData>
```

4. TRANSPORT ISSUES

4.1 Common RFC/Internet Protocols

Data formatted according to this specification is transferred using industry standard protocols

SunSpec Loggers and Hosts are required to at least support upload by HTTP/POST. Other transports, such as FTP may be specified in future versions.

4.2 HTTP Options

It is required that Hosts accept “Content-Encoding; gzip” for compression, although other compression methods may be supported as appropriate.

The Logger must use TCP Keepalives. The recommendation is to set this to less than 5 minutes to avoid problems with NAT’d connections and cellular links, where TCP sockets idle for more than 5 minutes tend to be flushed ungracefully without closing or aborting.

4.3 Security

Applications requiring security shall support SSL/TLS secure connections.

Hosts requiring authentication shall support at least HTTP Basic Authentication over SSL/TLS to allow login via a configured username and password.

Optionally, loggers and host can manage standard digital certificates (such as X.509) for strong authentication.

Under consideration for the next revision of the SunSpec Logger Protocol is adoption of the open, widely used standard OAuth as a more robust mechanism for authentication (see: http://oauth.net/). A strength of OAuth is that it does not require the surrender of credentials to a third-party service, a significant advantage given SunSpec's mission of improving interoperability. A lightweight implementation of OAuth most appropriate for embedded software is called two-legged OAuth 2.0. The workgroup welcomes comments from members on this subject.

5. HTTP Transport

This section specifies the format of the payload for HTTP REST data requests and responses

This interface provides the ability to send data elements to the server (host). A successful status means that the elements were accepted and persisted on the server. This interface does not support any status associated with the subsequent processing of the content of the data elements.

The XML schema (in appendix A) specifies the request and response elements. The following description summarizes the content of the schema.

5.1 Request

A data request contains one or more device records.

A device record MUST specify device identification, timestamp, and MAY specify a logger id and a result correlation id. A device record contains one or more model records. The timestamp provided in a device record is the default timestamp for all the points contained in the model record. A model record MUST specify a model id, and may specify in index.

A point MUST specify an id and MAY specify a unit, scale-factor, and a timestamp. A point contains a value associated with that point.

5.2 Response

A data response body can contain two levels of status information: a request level status, and a device record level status.

A server MUST provide a request level status and MAY provide a device record status if needed.

Request Status

The request status provides status for the request as a whole.

A request status MUST provide a status (SUCCESS, FAILURE, D_FAILURE) and MAY provide error information consisting of an additional code, error detail, and error message.

SUCCESS

A result of SUCCESS indicates that all of the device records were accepted without error by the server.

FAILURE

A result of FAILURE indicates that a request level failure occurred and the device records could not be processed. A FAILURE status MUST NOT be used to indicate a device record failure even if the request contains a single logger record.

If a device record failure occurs, it is necessary to use device record status to indicate the records that failed. The record correlation id is used to correlate the status element with the record element. If no correlation id is provided, device record status elements MUST be returned but they may not identify the record they are associated with.

D_FAILURE

A result of D_FAILURE indicates that some or all of the device records failed and a device record status is present for each device record that failed.

Device Record Status

A device record status provides the status for a single device record.

A device record status MAY provide error information consisting of an additional code, error detail, and error message.

5.3 HTTP REST Interface

The request/response encoding specified above does not depend on any specific transport.

It is intended that the service be provided as an HTTP REST style interface where an HTTP request is made to a service url and an HTTP response is returned.

Request

A request is sent as an HTTP POST with a body containing a single SunSpecData element.

Response

The response body contains a single SunSpecDataResponse. If the status in the request response is SUCCESS the HTTP status code MUST be 200.

If the status code in the request response is FAILURE, the HTTP status code MUST reflect the failure type.

If the status code in the request response is D_FAILURE, and there are multiple failure statuses in the status elements, then the HTTP status code MUST be 400. If there is a single failure status in the status elements, the HTTP status code MAY be 400 or MAY reflect the failure type.

The following HTTP status codes SHOULD be used for the listed failure status. It is not required that a Host support any or all of these status codes. For example, some Hosts may not return UNKNOWN_LOGGER error when a new logger begins reporting that has not yet been provisioned.

Status	HTTP Status Code	Summary/Reason
ACCESS_DENIED	403	Access granted to system but denied for specific resource
INVALID_MESSAGE	400	Message is either badly formed or has invalid content
INVALID_CREDENTIALS	401	Access to system denied
LIMIT_EXCEEDED	400	Limit for an amount within a single operation exceeded
PROCESSING_EXCEPTION	500	Error occurred during processing with a known reason
QUOTA_EXCEEDED	400	Limit for requests within a time range exceeded
SYSTEM_MAINTENANCE	503	System is currently unavailable
UNEXPECTED_EXCEPTION	500	Error occurred during processing for unknown reason
UNKNOWN_DEVICE	400	Device id is not valid
UNKNOWN_LOGGER	400	Logger id is not valid

Request Example with two devices, each a single model id.

This example includes two device models – one 3-phase inverter and one single-phase inverter. Note that this example uses the old V1 model point names.

By default, points which are not implemented should not be uploaded - although logger vendors may include an optional setting to force upload of all data, including those not implemented.

[UPDATE XXX]

```
<sunSpecData v=″1″>
 <d lid="11:22:33:44:55:66" id="GSC300-r300123456" man=″gsc″ mod=″r300″ sn="123456H"
t="2011-05-12T09:20:50Z" cid="1">
  <m id="103">
   <p id="A">30.43</p>
   <p id="AphA">10.01</p>
   <p id="AphB">10.22</p>
   <p id="AphC">10.20</p>
   <p id="PPVphAB">221.6</p>
   <p id="PPVphBC">222.5</p>
   <p id="PPVphCA">224.4</p>
   <p id="PhVphA">221.6</p>
   <p id="PhVPhB">222.5</p>
   <p id="PhVPhC">222.4</p>
   <p id="W">6701.3</p>
   <p id="Hz">60.01</p>
```

```
   <p id="WH">126973</p>
   <p id="DCA">14.28</p>
   <p id="DCV">469</p>
   <p id="DCW">6805</p>
   <p id="TmpCab">25.1</p>
   <p id="TmpSnk">45.6</p>
   <p id="TmpTrns">35.2</p>
   <p id="St">4</p>
   <p id="StVnd">14</p>
   <p id="Evt1">0</p>
  </m>
 </d>
 <d ns="mac" lid="11:22:33:44:55:66" man="gsc" mod="r100" sn="A123456"
t="2011-05-12T09:21:49Z" cid="2">
  <m id="101">
   <p id="A" sf="-2">3043</p>
   <p id="PhVphA" sf="-1">2216</p>
   <p id="W" sf="-1">67013</p>
   <p id="Hz" sf="-2">6001</p>
   <p id="WH">126973</p>
   <p id="DCA" sf="-2">1428</p>
   <p id="DCV">469</p>
   <p id="DCW">6805</p>
   <p id="TmpOt" sf="-2">3294</p>
   <p id="St">4</p>
  </m>
 </d>
</sunSpecData>
```

Request Example with a composite device, with 3 model ids

This example includes a 3-phase inverter with several environmental sensors. Notice the use of the ‘x’ attribute to index an array of similarly named data points.

```
<sunSpecData v="1">
 <d lid="11:22:33:44:55:66" id="GSC400-e101001" t="2011-05-12T09:20:50Z" cid="1">
  <m id="103">
   <p id="A">30.43</p>
   <p id="PhVphA">221.6</p>
   <p id="W">6701.3</p>
   <p id="WH">126973</p>
  </m>
  <m id="302">
   <p id="GHI" x="1">543</p>
   <p id="POAI" x="1">612</p>
   <p id="GHI" x="2">550</p>
   <p id="POAI" x="2">600</p>
   <p id="GHI" x="3">533</p>
   <p id="POAI" x="3">619</p>
  </m>
```

```
  <m id="303">
   <p id="TmpBOM" x="1">50.5</p>
   <p id="TmpBOM" x="2">60.1</p>
   <p id="TmpBOM" x="3">53.2</p>
  </m>
 </d>
</sunSpecData>
```

Request Example with a device with variable data

This example includes a string combiner with 3 configured input-strings. Notice the use of the 'x' attribute to index an array of string data.

```
<sunSpecData v="1">
  <d lid="11:22:33:44:55:66" man="gsc" mod="r100" sn="A123456" t="2011-05-
12T09:20:50Z" cid="AR45">
    <m id="401">
      <p id="DCA_Max">100</p>
      <p id="N">3</p>
      <p id="DCV">489</p>
      <p id="DCA" x="1">4.78</p>
      <p id="DCA" x="2">12.30</p>
      <p id="DCA" x="3">0.42</p>
    </m>
  </d>
</sunSpecData>
```

Request Example with an aggregate device with sub-devices indexed

This example includes an aggregate module system, where the modules are indexed, and so lack serial number and other detailed information.

```
<sunSpecData v="1">
  <d man="gsc" mod="r100" sn="A123456" t="2011-05-12T09:20:50Z">
    <m id="501">
      <p id="St">1</p>
      <p id="Evt">0</p>
    </m>
    <m id="502" x="1">
      <p id="St">1</p>
      <p id="Evt">0</p>
      <p id="Tms">123456789</p>
      <p id="OutW">678</p>
      <p id="Tmp">48.3</p>
    </m>
    <m id="502" x="2">
      <p id="St">1</p>
      <p id="Evt">0</p>
      <p id="Tms">123456795</p>
      <p id="OutW">634</p>
      <p id="Tmp">49.1</p>
    </m>
  </d>
</sunSpecData>
```

Request Example with an aggregate device with explicit sub-devices

This example includes an aggregate module system, where each module is treated as a device with serial number and other detailed information.

```
<sunSpecData v="1">
  <d man="gsc" mod="gw23" sn="A123456" t="2011-05-12T09:20:50Z">
    <m id="501">
      <p id="Stat">1</p>
      <p id="Evt">0</p>
    </m>
  </d>
  <d man="gsc" mod="md800" sn="Q123456" t="2011-05-12T09:20:50Z">
    <m id="502">
      <p id="Stat">1</p>
      <p id="Evt">0</p>
      <p id="Tms">123456789</p>
      <p id="OutW">678</p>
      <p id="Tmp">48.3</p>
    </m>
  </d>
  <d man="gsc" mod="md800" sn="Q234567" t="2011-05-12T09:20:50Z">
    <m id="502">
      <p id="Stat">1</p>
      <p id="Evt">0</p>
      <p id="Tms">123456795</p>
      <p id="OutW">634</p>
      <p id="Tmp">49.1</p>
    </m>
  </d>
</sunSpecData>
```

Request Good Response

Note: not all hosts will return the **sunSpecDataResponse** body, however all MUST return the standard HTTP status codes.

HTTP status: 200

response body:

```
<sunSpecDataResponse>
  <status>SUCCESS</status>
</sunSpecDataResponse>
```

Error Response Examples

This request includes a number of inverters, showing various error responses. Note that this logger uses duplicate cid for different device identities and hasn't included the logger id merely to make the example more concise.

```
<sunSpecData>
  <d id="GSC300-r300123456" t="2011-05-12T09:20:50Z" cid="1">
    <m id="101">
      <p id="A">30.43</p>
      <p id="PhVphA">221.6</p>
```

```
      <p id="W">6701.3</p>
      <p id="WH">126973</p>
    </m>
  </d>
  <d id="GSC300-r300123457" t="2011-05-12T09:21:50Z" cid="2">
    <m id="101">
      <p id="A">30.43</p>
      <p id="PhVphA">221.6</p>
      <p id="W">6701.3</p>
      <p id="WH">126973</p>
    </m>
  </d>
  <d id="GSC300-r300123458" t="2011-05-12T09:22:50Z" cid="1">
    <m id="101">
      <p id="A">30.43</p>
      <p id="PhVphA">221.6</p>
      <p id="W">6701.3</p>
      <p id="WH">126973</p>
    </m>
  </d>
  <d id="GSC300-r300123459" t="2011-05-12T09:23:50Z" cid="2">
    <m id="101">
      <p id="A">30.43</p>
      <p id="PhVphA">221.6</p>
      <p id="W">6701.3</p>
      <p id="WH">126973</p>
    </m>
  </d>
</sunSpecData>
```

Request level failure:

```
HTTP status: 503
```

Response body:

```
  <sunSpecDataResponse>
    <status>FAILURE</status>
    <code>SYSTEM_MAINTENANCE</code>
    <reason>Information about where the error was detected</reason>
    <message>User-friendly error information</message>
  </sunSpecDataResponse>
```

Single device record failure:

```
HTTP status: 403
```

Response body:

```
<sunSpecDataResponse>
  <status>D_FAILURE</status>
  <dResult id="GSC300-r300123456" t="2011-05-12T09:20:50Z" cid="1">
    <code>ACCESS_DENIED</code>
    <reason>Information about where the error was detected</reason>
    <message>User-friendly error information</message>
  </dResult>
</sunSpecDataResponse>
```

Multiple device record failure:

```
HTTP status: 400
```

Response body:

```
<sunSpecDataResponse>
  <status>D_FAILURE</status>
  <dResult id="GSC300-r300123456" t="2011-05-12T09:20:50Z" cid="1">
    <code>ACCESS_DENIED</code>
    <reason>Information about where the error was detected</reason>
    <message>User-friendly error information</message>
  </dResult>
  <dResult id="GSC300-r300123459" t="2011-05-12T09:23:50Z" cid="2">
    <code>UNEXPECTED_EXCEPTION</code>
    <reason>Information about where the error was detected</reason>
    <message>User-friendly error information</message>
  </dResult>
</sunSpecDataResponse>
```

About the SunSpec Alliance

The SunSpec Alliance is a trade alliance of developers, manufacturers, operators and service providers, together pursuing open information standards for the distributed energy industry. SunSpec standards address most operational aspects of PV, storage and other distributed energy power plants on the smart grid—including residential, commercial, and utility-scale systems—thus reducing cost, promoting innovation, and accelerating industry growth.

Over 70 organizations are members of the SunSpec Alliance, including global leaders from Asia, Europe, and North America. Membership is open to corporations, non-profits, and individuals. For more information about the SunSpec Alliance, or to download SunSpec specifications at no charge, please visit www.sunspec.org.