Status: Approved

Version: 1.1

SunSpec Modbus Client Conformance Test Procedures

SunSpec Specification

Abstract

This document specifies the conformance test procedures for compliance with the requirements specified in the *SunSpec Device Information Model Specification* and the associated specific SunSpec information model specifications.

License Agreement and Copyright Notice

Prepared by the SunSpec Alliance

Website: sunspec.org

Email: info@sunspec.org

Revision History

Version	Date	Comments
1.0	7-15-2024	Initial release
1.0.1	10-21-2025	Added an error test for unknown model ID present in the server
1.1	12-10-2025	Promoted from TEST to Approved.

About the SunSpec Alliance

The SunSpec Alliance is a California-based non-profit trade alliance that develops open information standards to support Distributed Energy Resource (DER) interoperability, cybersecurity, and grid resiliency. With over 180 member organizations from North America, Europe, Asia, Australia, and the Middle East, SunSpec serves manufacturers, software developers, utilities, and service providers across the DER industry.

SunSpec standards enable seamless integration of DER systems at residential, commercial, and utility scales, helping reduce cost, ensure compliance, and accelerate innovation. Membership is open to corporations, non-profits, and individuals.

About the SunSpec Specification Process

SunSpec Alliance specifications are initiated by SunSpec members 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 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 a SunSpec Interoperability Specification. These documents are 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 about best practices, but are not a matter of conformance. Informative documents can be revised more freely and more frequently to improve the quality and quantity of information provided.

SunSpec Interoperability Specifications follow a lifecycle pattern of: DRAFT, TEST, APPROVED, and SUPERSEDED.

For more information or to download a SunSpec Alliance specification, go to https://sunspec.org/about-sunspec-specifications/.

Table of Contents

1 Introduction 6

2 Test Procedures 6

2.1 Test Inputs 6

2.2 Test Categories 6

2.3 Server Setup 6

2.4 General Client Tests 7

2.4.1 CLI-1 – General Discovery for SunSpec Servers at Different IPv4 Addresses 8

2.4.2 CLI-2 – General Discovery for SunSpec Servers at Different Ports 8

2.4.3 CLI-3 – General Discovery for SunSpec Servers with Different Unit IDs 8

2.4.4 CLI-4 – General Discovery for SunSpec Servers with Different Starting Register Values 9

2.4.5 CLI-5 – General Discovery for SunSpec Servers with Different Baud Rates (Optional) 9

2.5 Read Tests 10

2.5.1 READ-1 – Single Point Reads 10

2.5.2 READ-2 – Multiple Point Reads 10

2.6 Write Tests 11

2.6.1 WR-1 – Write Single Point 11

2.6.2 WR-2 – Write Multiple Points 11

2.7 Information Tests 12

2.7.1 INFO-1 – SunSpec Type Interpretations 12

2.7.2 INFO-2 – Unimplemented Point Interpretations 13

2.8 Modbus Protocol Tests 13

2.8.1 PROT-1 – Partial Response 13

2.8.2 PROT-2 –TCP-Segmentation 13

2.9 Error Tests 14

2.9.1 ERR-1 – Noncompliant Server 14

2.9.2 ERR-2 – Exception Tests 14

2.9.3 ERR-3 – Unknown Model ID Test 15

1 Introduction

This document specifies the common SunSpec Modbus Client conformance test procedures.

2 Test Procedures

This section provides the test procedures for SunSpec Modbus Client compliance.

2.1 Test Inputs

SunSpec Modbus Client compliance testing is based on the functionality supported by the Client Under Test (CUT) device. The Client protocol implementation conformance statement (PICS) is used to specify the SunSpec Modbus functional content.

The device PICS is an Excel workbook that specifies the SunSpec models that are supported in the implementation. A SunSpec Modbus Client PICS template can be obtained from the SunSpec Alliance.

SunSpec standardized models are considered during certification testing. The vendor models that have been published in the SunSpec models repository should also be able to be tested. Any additional vendor-specific models must conform to SunSpec information modeling rules to the extent that they do not inhibit discovery and use of the standardized model supported in the device.

2.2 Test Categories

The SunSpec Modbus Client conformance tests fall into the following general categories:

- General client tests – Validates basic client functionality described in the PICS.
- Read tests – Validates the CUT can read from SunSpec Servers.
- Write tests - Validates the CUT can write to SunSpec Servers.
- Information tests - Validates the CUT understands the data represented in SunSpec Servers.
- Protocol tests - Validates the CUT handles the full protocol stack appropriately.
- Error tests - Validates the CUT interprets and logs communication errors.

2.3 Server Setup

Prior to testing, two servers will be configured by the test engineer with SunSpec Models and points that incorporate a range of models from the 100-series, 200-series, 700-series, and 800-series. Both Servers will include all the Models from the PICS document, in random order, with at least 5 other models injected at random in the model list. At least 1 of the models in the PICS will be duplicated. As an example, if the PICS included models 1 and 701-706, the following servers could be used for client testing.

Server 1	Server 2
Model 1	Model 1
Model 701	**Model 706**
Model 101	**Model 705**
Model 701	**Model 704**
Model 702	Model 203
Model 704	Model 213
Model 703	Model 12
Model 705	**Model 706**
Model 201	Model 802
Model 202	**Model 701**
Model 706	**Model 702**
Model 706	**Model 703**
Model 713	Model 803
Model 121	Model 803
Model 122	
Model 123	
Model 713	

Table 1 – Example Servers used for Client Conformance Tests

Additionally, the test engineer will capture all the data types present in all PICS models. For each data type, the test engineer will note if there is a SunSpec point with write functionality. The point types will include the following:

- int16
- uint16
- count
- acc16
- enum16
- bitfield16
- pad
- int32
- uint32
- acc32
- enum32
- bitfield32
- ipaddr
- int64
- uint64
- acc64
- ipv6addr
- float32
- float64
- string
- sunssf
- eui48

2.4 General Client Tests

SunSpec General Client Tests verify basic functionality for client devices. The following device tests must be performed.

Test	Description
CLI-1	General Discovery for SunSpec Servers at Different IP Addresses
CLI-2	General Discovery for SunSpec Servers at Different Ports
CLI-3	General Discovery for SunSpec Servers with Different Unit IDs
CLI-4	General Discovery for SunSpec Servers with Different Starting Register Values
CLI-5	General Discovery for SunSpec Servers with Different Baud Rates (Optional)

Table 2 - General Client Tests

2.4.1 CLI-1 – General Discovery for SunSpec Servers at Different IPv4 Addresses

This test validates the SunSpec Client Under Test (CUT) can interact with SunSpec Servers with different IP addresses.

2.4.1.1 Procedure

1. Run SunSpec Server 1 and SunSpec Server 2 on different IP addresses on an IPv4 network with the CUT equipment. One of the Servers may be run on localhost (address 127.0.0.1) on a system with the CUT software.
2. Share the server IP addresses, port numbers, and Unit IDs with the CUT operator or send this data to the CUT test interface.
3. Verify the CUT can connect to each Server, perform SunSpec model discovery, and extract all Common Model data.

2.4.1.2 Criteria

The Client SHALL accurately log all Models present in each Server and the contents of all the Common Model points. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

Network or serial traffic SHALL be analyzed to verify CUT operations.

2.4.2 CLI-2 – General Discovery for SunSpec Servers at Different Ports

This test validates the SunSpec Client Under Test (CUT) can interact with SunSpec Servers at different ports.

2.4.2.1 Procedure

1. Run SunSpec Server 1 and SunSpec Server 2 on different random ports between 1 and 65535. One or both servers may run on the localhost (address 127.0.0.1) on a system with the CUT software.
2. Share the server IP addresses, port numbers, and Unit IDs with the CUT operator or send this data to the CUT test interface.
3. Verify the CUT can connect to each Server, perform SunSpec model discovery, and extract all Common Model data.

2.4.2.2 Criteria

The Client SHALL accurately log all Models present in each Server and the contents of all the Common Model points. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

Network or serial traffic SHALL be analyzed to verify CUT operations.

2.4.3 CLI-3 – General Discovery for SunSpec Servers with Different Unit IDs

This test validates the SunSpec Client Under Test (CUT) can interact with SunSpec Servers with different Unit IDs.

2.4.3.1 Procedure

1. Run SunSpec Server 1 and SunSpec Server 2 with different Unit IDs between 1 and 247.

2. Share the server TCP settings (IP addresses, port numbers, and Unit IDs) or RTU settings (Baud rates and Unit IDs) with the CUT operator or send this data to the CUT test interface.
3. Verify the CUT can connect to each Server, perform SunSpec model discovery, and extract all Common Model data.

2.4.3.2 Criteria

The Client SHALL accurately log all Models present in each Server and the contents of all the Common Model points. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

Network or serial traffic SHALL be analyzed to verify CUT operations.

2.4.4 CLI-4 – General Discovery for SunSpec Servers with Different Starting Register Values

This test validates the SunSpec Client Under Test (CUT) can interact with SunSpec Servers with different staring register values.

2.4.4.1 Procedure

1. Run SunSpec Server 1 with the map starting at Modbus Register 0.
2. Share the server TCP or RTU settings with the CUT operator or send this data to the CUT test interface.
3. Verify the CUT can connect to the Server, perform SunSpec model discovery, and extract all Common Model data.
4. Run SunSpec Server 1 with the map starting at Modbus Register 40000.
5. Repeat step 2.
6. Run SunSpec Server 1 with the map starting at Modbus Register 50000.
7. Repeat step 2.

2.4.4.2 Criteria

The Client SHALL accurately log all Models present in each Server and the contents of all the Common Model points. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

Network or serial traffic SHALL be analyzed to verify CUT operations.

2.4.5 CLI-5 – General Discovery for SunSpec Servers with Different Baud Rates (Optional)

This test validates the SunSpec Client Under Test (CUT) can interact with SunSpec Servers with different baud rates.

2.4.5.1 Procedure

1. Run SunSpec Server 1 with minimum supported baud rate from the CUT PICS.
2. Share the server RTU settings with the CUT operator or send this data to the CUT test interface.
3. Verify the CUT can connect to the Server, perform SunSpec model discovery, and extract all Common Model data.
4. Rerun Steps 2-3 for each subsequent supported baud rate included in the CUT PICS.

2.4.5.2 Criteria

The Client SHALL accurately log all Models present in each Server and the contents of all the Common Model points. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

Network or serial traffic SHALL be analyzed to verify CUT operations.

2.5 Read Tests

Test	Description
READ-1	Single Point Reads
READ-2	Multiple Point Reads

Table 3 - Read Tests

2.5.1 READ-1 – Single Point Reads

This test validates that the SunSpec Client Under Test (CUT) can read SunSpec Servers points one-by-one.

2.5.1.1 Procedure

1. Connect the CUT to Server 1.
2. Verify all points in the Common model (model 1) can be read individually.
3. Repeat Step 2 for each of the models in the PICS.

2.5.1.2 Criteria

The Client SHALL accurately log all points for all PICS models. The log data SHALL be represented as hex strings. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

Network or serial traffic SHALL be analyzed to verify CUT operations.

2.5.2 READ-2 – Multiple Point Reads

This test validates the SunSpec Client Under Test (CUT) can read SunSpec Servers points at lengths up to the Modbus maximum read length (125 registers).

2.5.2.1 Procedure

1. Connect the CUT to Server 1.
2. Verify all points after the ID and L points in the Common model (model 1) SHALL be read with a single request.
3. Repeat Step 2 for each of the models in the PICS. If the model length exceeds 125 registers, the model SHALL be read in multiple reads that maximize the number of points read in each holding register read operation. Partial point reads are not required.

NOTE: this test may be completed in the discovery process in CLI-1, CLI-2, CLI-3, or CLI-4.

2.5.2.2 Criteria

The Client SHALL accurately log all points for all PICS models. The log data SHALL be represented as hex strings. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

Network or serial traffic SHALL be analyzed to verify CUT operations.

2.6 Write Tests

Test	Description
WR-1	Write Single Point
WR-2	Write Multiple Points

Table 4 – Write Tests

2.6.1 WR-1 – Write Single Point

This test validates that all implemented adjustable points in the model can be written individually using Modbus Function Code 0x06.

2.6.1.1 Procedure

1. Verify all implemented adjustable points can be written to the minimum value, maximum value, and three intermediate values as defined in the Server 1 PICS using Modbus function code 0x06. If an adjustable point has fewer possible values, all possible values must be tested.
   - This can be done by providing the Server 1 PICS to the CUT operator or programmatically sending the CUT the 5 values for each point.
2. For adjustable enumerated points, verify each supported value in the Server 1 PICS can be written using Modbus function code 0x06.
3. If the CUT supports RTU server interfaces as indicated in the PICS, rerun Steps 1 and 2 using Unit ID = 0 (the broadcast ID).

2.6.1.2 Criteria

The Client SHALL accurately log all write operations. The log data SHALL be represented as hex strings. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

Server write operations SHALL be validated by the test engineer.

Network or serial traffic SHALL be analyzed to verify CUT operations.

2.6.2 WR-2 – Write Multiple Points

This test validates that all implemented adjustable points in the model can be written using Modbus Function Code 0x10. This test does not require that multiple points are written at once, just that the 0x10 functions code can be used to write each of the points.

2.6.2.1 Procedure

1. Verify all implemented adjustable points can be written to the minimum value, maximum value, and three intermediate values as defined in the Server 1 PICS using Modbus function code 0x10. If an adjustable point has fewer possible values, all possible values must be tested.
   - This can be done by providing the Server 1 PICS to the CUT operator or programmatically sending the CUT the 5 values for each point.
2. For adjustable enumerated points, verify each supported value in the Server 1 PICS can be written using Modbus function code 0x10.

3. If the CUT supports RTU server interfaces as indicated in the PICS, rerun Steps 1 and 2 using Unit ID = 0 (the broadcast ID).

2.6.2.2 Criteria

The Client SHALL accurately log all write operations. The log data SHALL be represented as hex strings. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

Server write operations SHALL be validated by the test engineer.

Network or serial traffic SHALL be analyzed to verify CUT operations.

2.7 Information Tests

The following tests must be performed for each SunSpec supported standardized model specified in the PICS.

Test	Description
INFO-1	SunSpec Type Interpretations
INFO-2	Unimplemented Point Interpretations

Table 5 - Information Tests

2.7.1 INFO-1 – SunSpec Type Interpretations

This test validates that the CUT can interpret all data points in the PICS.

2.7.1.1 Procedure

1. Connect the CUT to Server 1.
2. Have the CUT read all Common model (model 1) points and print the human-readable information for these points. The values for the points SHALL comply with the following:
   - The CUT SHALL log the decimal value for all numerical points (*int16*, *int32*, *int64*, *uint16*, *uint32*, *uint64*), and convert all points using the appropriate scale factors.
   - The CUT SHALL log the decimal value for all points with floating points (*float32*, *float64*).
   - The CUT SHALL log *ipaddr* points in dotted-decimal format, e.g., 192.168.1.2.
   - The CUT SHALL log *ipv6addr* points with 128 bits written in hexadecimal notation, e.g., 2001:db8:1234::f350:2256:f3dd. The leading zeros in each group can be removed and two or more consecutive groups of zeros can be replaced by a double colon (::).
   - The CUT SHALL log *eui48* points in capitalized decimal-hex format, e.g., B8:27:EB:12:34:56.
   - The CUT SHALL log *bitfield16* and *bitfield32* points as hex strings.
   - The CUT SHALL log *string* as human-readable UTF-8 strings.
   - The CUT SHALL log *enum16* and *enum32* points as the base 10 value (not name).
   - The CUT SHALL log all other points (*pad*, *count*, *sunssf, acc16*, *acc32*, *acc64)* as base 10 values.
3. Repeat Step 2 for each of the models in the PICS. Duplicated models may be skipped,

i.e., the client may skip additional models of the same model ID.

2.7.1.2 Criteria

The Client SHALL accurately log all values in human-readable values. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

2.7.2 INFO-2 – Unimplemented Point Interpretations

This test validates that the CUT can interpret unimplemented points.

2.7.2.1 Procedure

1. Connect the CUT to Server 1.
2. For all the data types present in the PICS models, update Server 1 to configure at least one of these points in the PICS models to the unimplemented value.
3. Have the CUT read all SunSpec points in all models in Server 1 and log the unimplemented points.

2.7.2.2 Criteria

The Client SHALL accurately log all unimplemented values in the PICS models. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

2.8 Modbus Protocol Tests

Test	Description
PROT-1	Partial Response
PROT-2	TCP Segmentation (Only for Modbus TCP Clients)

Table 6 – Protocol Tests

2.8.1 PROT-1 – Partial Response

This test validates the CUT can recover from an incomplete Modbus response message.

2.8.1.1 Procedure

1. Preconfigure Server 1 to respond with a partial Modbus response.
2. Connect CUT to Server 1.
3. Send a Common model read request from the client.
4. Reconfigure Server 1 to behave normally.
5. Connect CUT to Server 1.
6. Send a Common model read request from the client.
7. Verify the CUT logs the read values as expected.

2.8.1.2 Criteria

The Client SHALL remain functional after receiving the partial response. The read log in Step 7 may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

2.8.2 PROT-2 –TCP-Segmentation

This test validates the CUT can support a Modbus response divided across multiple TCP frames.

2.8.2.1 Procedure

1. Preconfigure Server 1 to respond with a TCP-segmented Modbus response.
2. Connect CUT to Server 1.
3. Send a read request from the CUT.
4. Verify the CUT can parse the data from the TCP-segmented Modbus response.

2.8.2.2 Criteria

The Client SHALL accurately log all unimplemented values in the PICS models. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

2.9 Error Tests

These tests are performed to verify that the device under test performs exception handling correctly.

Test	Description
ERR-1	Noncompliant Server
ERR-2	Exception Tests
ERR-3	Unknown Model ID

Table 7 – Error Tests

2.9.1 ERR-1 – Noncompliant Server

This test verifies the CUT recovers when connecting to a non-SunSpec compliant Server.

2.9.1.1 Procedure

1. Configure Server 1 to start the SunSpec registers at holding register 40001.
2. Attempt to connect the CUT to Server 1.

2.9.1.2 Criteria

The Client SHALL accurately log the noncompliant server. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

2.9.2 ERR-2 – Exception Tests

This test verifies the CUT appropriately interprets, logs, manages, and recovers from Modbus Exceptions.

2.9.2.1 Procedure

1. Determine operations for Server 1 that generate Exception codes 1, 2, 3 and 4. Examples include:
    a. Requesting illegal function code 0x88
    b. Writing a Read-Only register
    c. Writing to unimplemented point
    d. Writing an Enum register to a value that is not defined or supported in the Server.

2. Request operations from Server 1 that produces Exception codes 1, 2, 3 and 4.
3. Verify the CUT logs each Modbus Exception and continues to operate normally.

2.9.2.2 Criteria

The Client SHALL accurately log all exception codes. The log may be displayed to the screen, saved to a file, transferred to the testing software, or otherwise shared with the test engineer.

2.9.3 ERR-3 – Unknown Model ID Test

This test verifies the CUT appropriately responds to presence of models with unknown IDs.

2.9.3.1 Procedure

1. Configure Server 1 to start the SunSpec map that includes a model with a model ID that is not in the CUTs model definition directory.

2. Attempt to connect the CUT to Server 1.

2.9.3.2 Criteria

The Client SHALL connect to the server without issues. The list of discovered models MUST not include the model with the unknown ID.