Open Source DER Outstation for DNP Application Note AN2018-001

Reference Implementation of DNP Application Note AN2018-001 – "DNP3 Profile for Communications with Distributed Energy Resources"

3002015355

Open Source DER Outstation for DNP Application Note AN2018-001

3002015355

Software Manual, February 2019

EPRI Project Managers

B. Ealey

ELECTRIC POWER RESEARCH INSTITUTE
3420 Hillview Avenue, Palo Alto, California 94304-1338 ▪ PO Box 10412, Palo Alto, California 94303-0813 ▪ USA
800.313.3774 ▪ 650.855.2121 ▪ askepri@epri.com ▪ www.epri.com

THE ELECTRIC POWER RESEARCH INSTITUTE (EPRI) PREPARED THIS REPORT

This is an EPRI Technical Update report. A Technical Update report is intended as an informal report of continuing research, a meeting, or a topical study. It is not a final EPRI technical report.

NOTE

For further information about EPRI, call the EPRI Customer Assistance Center at 800.313.3774 or e-mail askepri@epri.com.

ACKNOWLEDGMENTS

The Electric Power Research Institute (EPRI) prepared this report.

Principal Investigators

J. Anandan
B. Ealey

This report describes research sponsored by EPRI.

This publication is a corporate document that should be cited in the literature in the following manner:

Open Source DER Outstation for DNP Application Note AN2018-001: Reference Implementation of DNP Application Note AN2018-001 – "DNP3 Profile for Communications with Distributed Energy Resources". EPRI, Palo Alto, CA: 2019. 3002015355.

SOFTWARE DESCRIPTION

This report summarizes EPRI's work to develop an open source implementation of an outstation supporting DNP Application Note AN2018-001 – DNP3 Profile for Communications with Distributed Energy Resource.

Description

The DNP3 Outstation stack implements DNP Application Note AN2018-001 – DNP3 Profile for Communications with Distributed Energy Resources (DERs). It simplifies the process of adding DNP3 to DERs by providing developers with a simple interface that they can connect to their software. The open source software absorbs the complexities of DNP Application Note AN2018-001 by performing internal processes that are invisible to the user so that the DER follows the application note without having to actively implement those requirements in the product. Developers can then use the code directly in their products or use it to help guide their own development.

Benefits and Value

This software benefits developers looking to implement DNP Application Note AN2018-001 in smart inverter (solar or storage). The software provides the following benefits:

- Simplifies implementation of the application note by absorbing its complexities.
- Vetted by a third-party through conformance testing.
- It was built on existing, well known open source libraries and supports a suite of features needed to simplify development.

Platform Requirements

The software was developed using C++ and was tested in a Linux environment.

Keywords

DNP3
Smart Inverters
Interoperability
Communications
Open Source

LEGAL NOTICE

The development of this software is funded by the CEC through the Electric Program Investment Charge (EPC-15-089), which funds clean energy research, demonstration, and deployment projects that support California's energy policy goals and promote enhanced reliability and safety, at lower costs.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

MANDATORY SOFTWARE INSTALLATION INFORMATION

Both software statements below (and their titles) need to be inserted into the Installation section of the software manual.

Installation of EPRI Software at Client Site

This software uses third party software products, operating systems, and hardware platforms. Over time, security issues may be uncovered in these third party products. You should review your use of this software with your Information Technology (IT) department to ensure that all recommended security updates and patches are installed to all third party products when needed.

If you experience difficulties accessing the application

If you experience difficulties accessing the application after standard installation on Windows 7, Windows 8.1 or Windows 10, please consult your IT department personnel to have proper access permissions setup for your use. If the problem cannot be resolved, please call the EPRI Customer Assistance Center (CAC) at 1-800-313-3774 (or email askepri@epri.com).

1
INTRODUCTION

Utilities worldwide are challenged by the technical and business decisions to embrace distributed energy resources (DER) as a key part to support a flexible, reliable, integrated, and renewable grid. DER technologies include demand response (DR) of controllable loads, distributed generation (DG), energy storage, and electric vehicles (EV). The design and implementation for robust and interoperable communications between DER technologies and other grid systems are at the heart of addressing many of these challenges. Communications can enable advanced capabilities needed to support a reliable and secure grid including dynamic monitoring and control of diverse DER[1,2,3]; system analytics to improve efficiency and effectiveness; and optimized dispatch of DER to improve use.[4,5]

Electric utilities across the world are looking at methods to operate the distribution system more effectively in alignment with the increase of DER technologies such as distributed solar photovoltaic (PV), energy storage, and controllable loads for demand response. These methods include embracing direct[6,7], non-proprietary access across a variety of vendor makes and models of DER; optimizing the dispatch of DER through advanced distribution grid management systems including Distributed Energy Resourced Management Systems (DERMS) and Advanced Distribution Management Systems (ADMS); and enabling data analytics for grid modeling to unlock new undiscovered use cases. Though these future states are long-term goals, there are utility projects and vendors across the world looking to make these systems a reality through

---

1 Common Functions for Smart Inverters: 4th Edition. EPRI, Palo Alto, CA: 2016. 3002008217.

2 Common Demand Response Functions for Heating, Ventilating, and Air Conditioning (HVAC): A Summary of Demand Response Functionality Discussed in the Industry to Date. EPRI, Palo Alto, CA: 2017. 3002011045.

3 Applying Standards-Based Demand Response to Support Solar Integration: A Summary of EPRI Testing at the National Renewable Energy Laboratory (NREL). EPRI, Palo Alto, CA: 2017. 3002009849.

4 DER Grouping Methods and Considerations for Operations: A Study on the Different Approaches for Creating and Managing Groups of DER and the Impact on Operations. EPRI, Palo Alto, CA: 2017. 3002009857.

5 Common Functions for DER Group Management, Third Edition. EPRI, Palo Alto, CA: 2016. 3002008215.

6 “Direct Access” relates to where in the communication pathway an open, standard interface exists at the device.

7 The Value of Direct Access to Connected Devices. EPRI, Palo Alto, CA: 2017. 3002007825

pilots and demonstrations.[8,9,10,11] Under this not-so-distant future state is a communications backbone that supports standardized and direct access to DER devices and grid systems and allows access to the DER over the long-term use of the system.[7] In response to these challenges; utilities, regulators, industry, and trade organizations are planning to add communications requirements in their specifications, mandates, and proposal criteria that support their short-term and long-term grid planning and operation, policy, and business needs.[12,13] However, for this to be effective the standards must support the capabilities of those systems so their benefit can be fully realized.

As the grid integrates more DER, communications standards become essential for ensuring that utilities and the public can maximize the operational functionality and financial value of distributed and traditional grid assets. Significant work to date has addressed communication standards for smart inverter DER systems in general, but interoperable communication standards to support large-scale energy storage systems (ESS) are still in development. At the time this project started, current standards and protocols were based on earlier global efforts to define standard functionality for smart inverters. However, the primary focus of early-stage implementation of communication standards and protocols by manufacturers and in evaluations of field demonstrations have been on solar photovoltaic (PV) systems with limited storage system deployments. These solar PV profiles include energy storage device- and system-level functionality but there is a need to expand and refine the function set with energy storage in mind.

---

[8] Southern California Edison, "Grid Modernization Initiative: Grid Management System Architecture," February 2016, http://www.edison.com/content/dam/eix/documents /innovation/SCE%20Grid%20Management%20System%20Architecture%202.1.16b.pdf. Accessed March 2019

[9] Shuichi Ashidate. Tokyo Electric Power Co., Inc. "Opportunities and Challenges for Smart Grid in Japan". March 8, 2016. http://www.nedo.go.jp/content/100778194.pdf. Accessed March 2019

[10] Beneficial Integration of Energy Storage and Load Management with Photovoltaics. Office of Energy Efficiency and Renewable Energy. U.S. Department of Energy. https://www.energy.gov/eere/solar/project-profile-electric-power-research-institute-shines. Accessed March 2019

[11] Tucson Electric Power Project RAIN. EPRI, Palo Alto, CA: 2018. 3002014812

[12] Interoperability Strategic Vision – A GMLC White Paper. Pacific Northwest National Laboratory. March 2018. https://gridmod.labworks.org/sites/default/files/resources/ InteropStrategicVisionPaper2018-03-29.pdf. Accessed March 2019

[13] IEEE Standard 1547™ — Communications and Interoperability: New Requirements Mandate Open Communications Interface and Interoperability for Distributed Energy Resources. EPRI, Palo Alto, CA: 2017. 3002011591

Project Overview

The open source software discussed in this report is one of component of a larger project. The project improves the ability of grid operators to communicate with distributed energy resources (DER) and enhance use and value of energy storage and solar generation connected to smart inverters. It addresses this need by building on existing work and accelerating the dialogue and support for open communication with energy storage systems. It focuses on DNP3 for smart inverters, which is a standard commonly used to communicate with large, distribution-system connected energy storage systems and other distribution assets.

A four-step process was used:

1. The team created an open, collaborative working group to perform a gap analysis to evaluate current communications standards and identify opportunities that can help the realize the full set of benefits from solar and storage systems.
2. The team applied the lessons learned and work with the DNP3 Users Group to publish an update to the existing energy storage and solar specifications application note so it will support the most advanced energy storage use cases – DNP Application Note AN2018-001[14]. The team also shared the lessons learned with other communication standards so that they could benefit from the findings.
3. The team developed open-source DNP3 software to simplify product development for manufacturers or other industry stakeholders. It was built on existing, well known open source libraries and supports a suite of features needed to simplify development.
4. The team developed a framework for conformance testing framework to ensure interoperability. It includes (1) the DNP3 Process Logic which defines the test procedure and pass/fail criteria, (2) the Protocol Information Conformance Statement (PICS) which lists which parameters are supported or unsupported by the product under test to inform which tests need be performed on the distributed energy resource, and (3) programming an existing test software suite to automate testing and validation of pass/fail criteria.

This report focuses on (3) open-source DNP3 software to implement DNP Application Note AN2018-001 – "DNP3 Profile for Communications with Distributed Energy Resources".

Purpose of Open Source DNP3 Software

When new standards or procedures are published, and the industry must develop capabilities to support the requirements in products or systems. The team built an open-source outstation to reduce the time and effort required for vendors and other stakeholders to integrate DNP3 into smart inverters (energy storage or solar) and other devices. The open source software has been tested using the framework to validate standards conformance that was developed in parallel to verify its conformance to the application note.

---

[14] DNP Users Group, DNP3 Application Note AN2018-001 - DNP3 Profile for Communications with Distributed Energy Resources. January 2019, https://www.dnp.org/LinkClick.aspx?fileticket=BiDPDkwxy70%3d&portalid=0&timestamp=1553622205077. Accessed March 2019.

This goal of the open source software is to advance communication capabilities of solar and energy storage systems by providing the necessary tools to implement the protocol with minimal understanding of the technical details of the application note. For example, a manufacturer could use the open source DNP3 software in their product to ensure that their product will be interoperable with other DNP3 devices that also follow the application note.

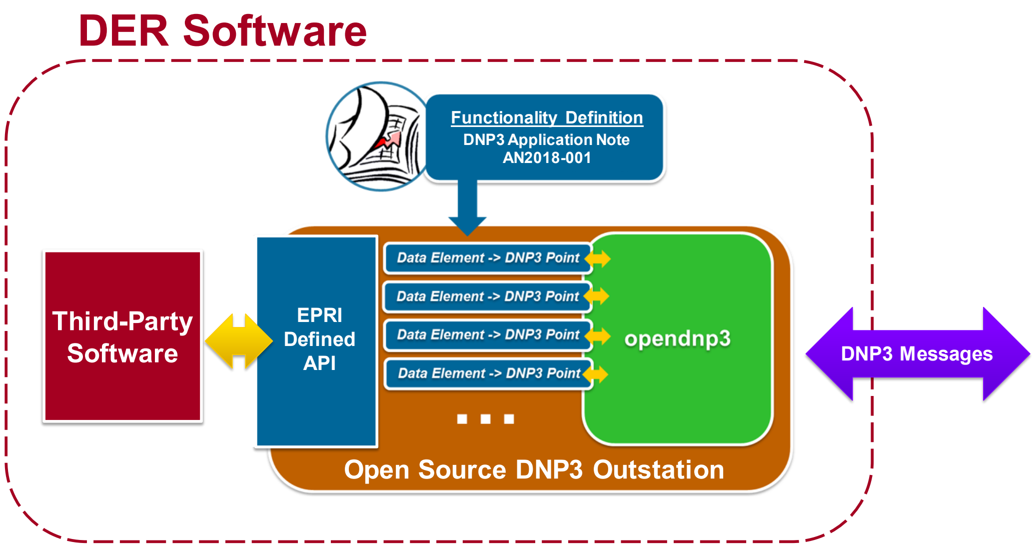

Figure 1-1
High-Level Architecture of Open Source DNP3 Outstation

The software will not stand on its own. Two other components are needed for the system to connect to a DNP3 network and exchange data and control parameters.

(1) Downstream applications / third-party software (e.g. manufacturer software and hardware). The open source software must be integrated with a manufacturer's system to implement the protocol. The downstream application collects data from the open source code as new control parameters are sent to the distributed energy resource. It will also update the open source code with updated monitoring points.

(2) opendnp3[15]. The project leveraged existing open source software to help keep the project economical. opendnp3 was used to implement the underlying, generic DNP3 functionality.

Development of Open Source DNP3 Software

The open source DNP3 software implements an outstation following DNP Application Note AN2018-001 – DNP3 Profile for Communications with Distributed Energy Resources. It simplifies the process of adding DNP3 to DERs by providing developers with a simple interface that they can connect to their software. The open source software absorbs the complexities of DNP Application Note AN2018-001 by performing internal processes that are invisible to the user so that the DER follows the application note without having to actively implement those

[15] https://www.automatak.com/opendnp3/

requirements in the product. Developers can then use the code directly in their products or use it to help guide their own development.

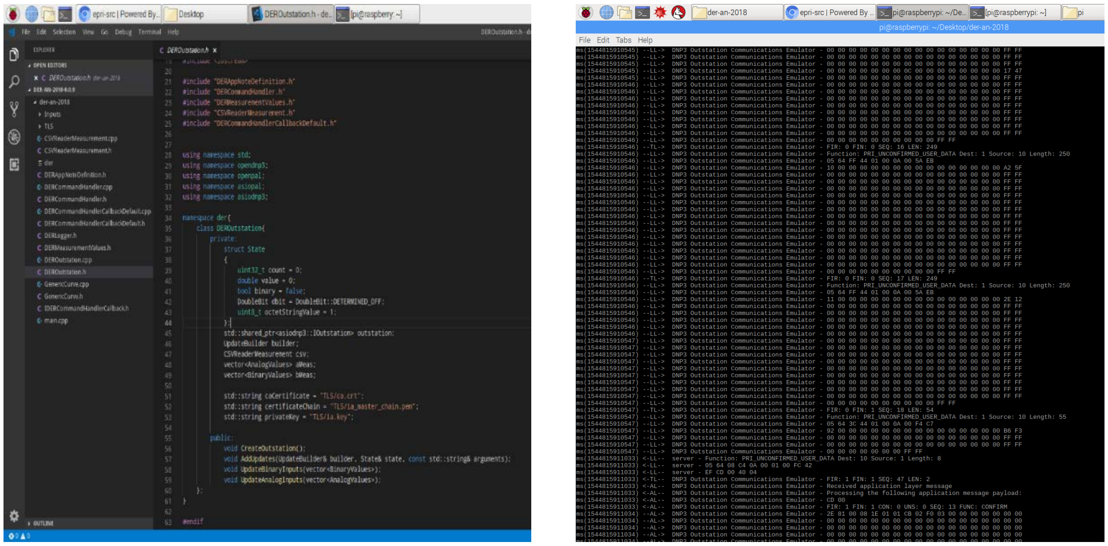

Figure 1-2
Open Source Outstation Running in Terminal Window

The open source software was developed in C++ on a Linux based operating system. Testing was performed on a Raspberry Pi 3. The code uses Automatak's opendnp3 library to implement the underlying DNP3 requirements. The team's contribution included adding the DER specific aspects defined in DNP Application Note AN2018-001. It uses OpenSSL to implement Transport Layer Security (TLS).

Figure 1-3
Raspberry Pi 3

The following smart inverter functions are supported in the open source software. Not all functions and parameters defined in Application Note AN2018-001 are included in the open source software however examples are provided for each type of function and command, so developers can easily copy the format to add new functions.

- Enter Service
- Active Power Limit Mode
- Constant Power Factor Mode
- Volt-VAR Mode
- Watt-VAR Mode
- Constant VAR Mode

- Volt-Watt Mode
- Voltage Trip
- Momentary Cessation
- Frequency Trip
- Frequency Droop (Freq-Watt)
- Monitoring Points
- *Scheduling is implemented for all functions above.*

Conformance Testing of Open Source Software

The Open Source DNP3 Outstation was tested by the framework for conformance testing which was created in parallel within the scope of this project. The SunSpec Alliance performed the testing and provided certificate[16] to show the tests performed and results.

The conformance process is designed to verify requirements defined in Application Node AN2018-001. This includes validating that the outstation is…

…able to obtain all required input points associated with tested functionality
…able to update all required output points associated with tested functionality
…validate that input points track output points perform correctly
…validate generic curve management functionality
…validate points are implemented or marked as unimplemented consistent with the logical point groups in which they reside.

The conformance testing process does not validate that the outstation…

…follows underlying DNP3 requirements. This is addressed by existing processes.
…responds with the correct electric response or that monitoring points are accurate. This is handled by functional test standard like UL-1741SA and IEEE 1547-2018.

SUNSPEC CERTIFIED

Certification Details for DNP3-AN2018

Key	Value
Certificate Type	IEEE 1815/AN2018
Certificate Number	REF000000001
Company Name	Electric Power Research Institute
Company Address	3420 Hillview Ave
Company City	Palo Alto
Company State/Province	CA
Company Country	USA
Company Postal Code	94304
Date Issued	01/15/2019
Test Laboratory	Not Applicable
Supervising Test Engineer	Bob Fox
Certificate Signer Name	Tom Tansy
Software Name 1	der-an-2018-0.1.5
Software Version 1	0.1.5
Software Checksum 1	e3c794330a011441331078a1b59f9b34
Operating System 1	Raspbian
Operating System Version 1	8 (Jessie)
Software Operating Environment	Device
Protocol Implementation Conformance Statement	DNP3-AN2018-PICS-EPRI.xlsx
Cloud Provider	Not Applicable
Cloud Provider Version	Not Applicable
Hardware Manufacturer	Raspberry Pi
Hardware Model 1	Raspberry Pi 3
Test Completion Date	01/15/2019
Test Description	Test performed in compliance with DNP Application Note AN2018-001 using test procedures specified in DNP DER Application Note Test Procedures.
Additional Test Comments	This evaluation report is of a reference implementation that was modified in order to verify the functionality of the protocol stack. The evaluation was not performed in a SunSpec Authorized Testing Laboratory and therefore no official certification can be provided.
Test MON-001	PASS

SUNSPEC CERTIFIED

Test ALARM-001	PASS
Test CONN-001	NOT SUPPORTED
Test SERV-001	PASS
Test OP-001	PASS
Test CURVE-001	PASS
Test CURVE-002	PASS
Test CURVE-003	PASS
Test VRT-001	PASS
Test FRT-001	PASS
Test FW-001	PASS
Test DRCS-001	NOT SUPPORTED
Test DVW-001	NOT SUPPORTED
Test APL-001	PASS
Test CHG-001	NOT SUPPORTED
Test CCM-001	NOT SUPPORTED
Test APR1-001	NOT SUPPORTED
Test APR2-001	NOT SUPPORTED
Test APR3-001	NOT SUPPORTED
Test AGC-001	NOT SUPPORTED
Test APS-001	NOT SUPPORTED
Test VW-001	PASS
Test FWC-001	NOT SUPPORTED
Test CVAR-001	PASS
Test FPF-001	PASS
Test VV-001	PASS
Test WV-001	PASS
Test PFC-001	NOT SUPPORTED
Test PSIG-001	NOT SUPPORTED
Test SCHED-001	PASS

Figure 1-4
Certification Details for Open Source DNP3 Outstation

---

[16] The certificate was modified slightly to acknowledge this was a reference implementation.

Ongoing Developments for DER Software

EPRI has produced numerous tools through their research projects. Each year EPRI produces a report summarizing these tools called *EPRI's DER Integration Toolkit: An Overview of EPRI Tools for Testing and Implementing Open Protocols*[17]. The DER Integration Toolkit report pulls these tools together into a repository of test tools and implementation resources for applying open communication protocols to both demand response and distributed energy resources applications. The goal of the toolkit is to help support development and testability of open protocols. The end goal is to create a "demonstration in a box" or "in-the-loop" testing where any component of the communication architecture can be simulated or implemented using components of EPRI's DER Integration Toolkit.

The toolkit expands each year. The industry continues to improve current communications standards to meet modern use cases for the industry continues to improve current communications standards to meet modern use cases for solar, energy storage, and demand response systems. As the industry evolves EPRI plans to adapt their DER Integration Toolkit, including the Distributed Energy Resources Simulator, to meet the needs of the industry.

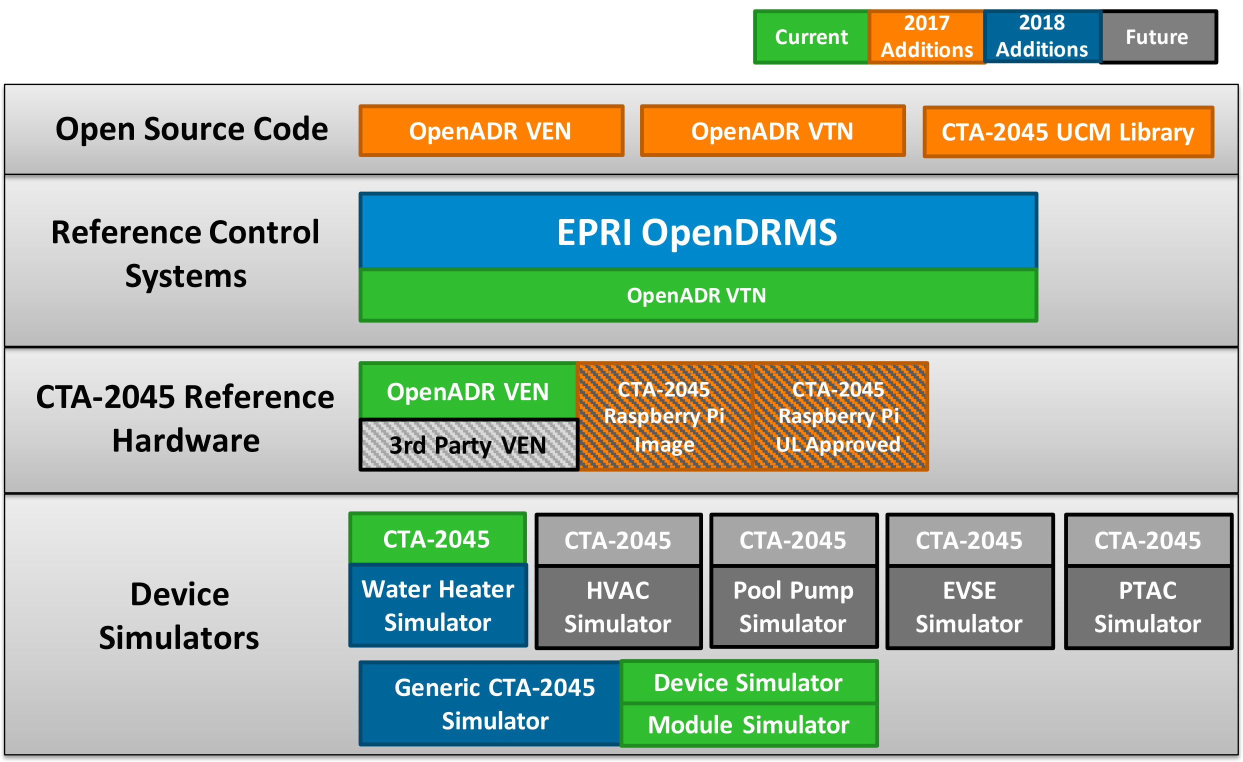

Figure 1-5
Upcoming Tools for Demand Response Technologies.

---

[17] *EPRI's DER Integration Toolkit: An Overview of EPRI Tools for Testing and Implementing Open Protocols.* EPRI, Palo Alto, CA: 2018. 3002013623.

In the immediate term EPRI plans to continue to expand the device simulators to support additional device types including energy storage, heating and cooling systems, pool pumps, electric vehicles, and commercial room air conditioners. EPRI is also looking to expand protocol support across the entire toolkit to include all relevant communications protocols. The reference control systems and device simulators have been intentionally designed to scale well to include new protocols as needs arise. EPRI is funded through base research programs, supplemental projects with utilities, and government funded activities to expand the tools in the DER toolkit.

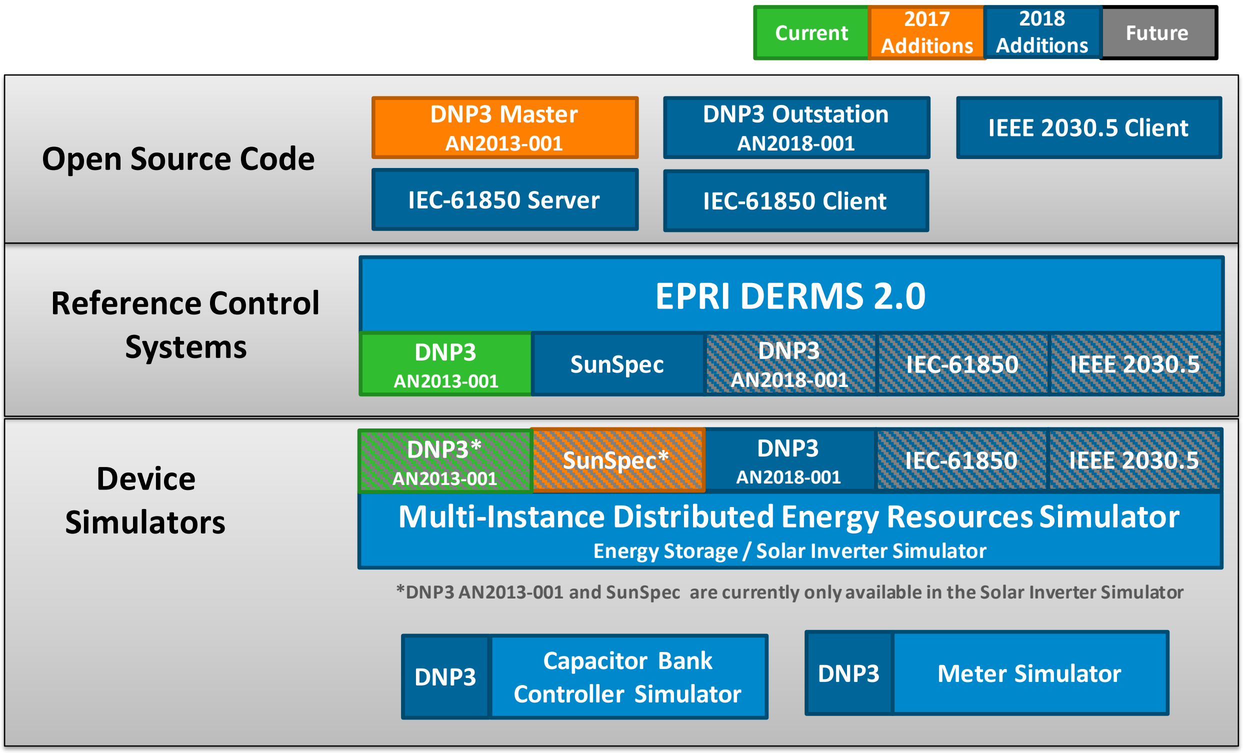

Figure 1-6
Upcoming Tools for Inverter-based Technologies.

In addition to adding new tools EPRI is constantly receiving feedback from users of the toolkit including feature requests and bugs. This feedback is reviewed and included in the work plan for each tool. Severe bugs are addressed immediately.

EPRI is working with Sandia National Laboratory in designing a distributed monitoring system that can patrol a wide range of cyber-attack vectors, detect various attack methods, predict adversary movements, and implement controls that mitigate damage to DER devices. As part of this project, EPRI will update the DER Smart functions to latest IEEE 1547 functions. EPRI will add Transport Layer Security (TLS) to the Modbus protocol driver. EPRI is also working with EDF Energy on integrating IEC 61850 drivers to OpenDERMS and the device simulators in the next year[18]. Another development work planned for 2019 is an IEEE 2030.5 to SunSpec Modbus translator. To meet the recent requirements of Rule – 21 and IEEE 1547, there is much interest

[18] *IEC 61850 Protocol Driver Agents: Open Source Software to Support Testing and Development of IEC-61850 in Smart Inverters*. EPRI, Palo Alto, CA: 2018. 3002013625.

among DER vendors to develop an IEEE 2030.5 to Modbus translator. EPRI will extend the capability of the currently available, open source IEEE 2030.5 client, to a translator that can communicate with SunSpec Modbus compliant devices. As part of this project, a specification that documents the mapping between the two communication protocols will also be developed.

EPRI is committed to the development of test tools to ease the entry of open protocols into the market and plans to continue to develop, maintain, and support these tools for members. The end goal is to create a “demonstration in a box” where any component of the communication architecture can be validated, tested, or implemented using components of the DER Integration Toolkit. More information on other tools in EPRI’s DER Integration Toolkit can be found in the toolkit summary report.

2
BUILDING THE DER OUTSTATION

Platform and prerequisites

1) The targeted operating system is Linux. System was developed and tested on Raspbian (default OS for Raspberry Pi).
2) System was developed and tested on the Raspberry Pi 3.
3) GCC compiler 4.9.2 or greater.

Installation of Raspbian OS in VirtualBox

All the developments and debugging for the DER Outstation was done on Raspbian x86 installed on a virtual machine hosted on a windows machine. VirtualBox was chosen as the hypervisor because it is free, open-source, and simple to install and configure. Raspbian and VirtualBox can be installed by following the instructions at:

http://www.aoakley.com/articles/2017-07-04-raspbian-x86-virtualbox.php

Installation of Raspbian OS in Raspberry Pi

All the testing of the DER Outstation was done on Raspbian ARM installed on a Raspberry Pi 3. NOOBS, an operating system installation manager was used to install Raspbian on Raspberry Pi 3. Raspbian can be installed on the Pi by following the instructions at:

https://projects.raspberrypi.org/en/projects/noobs-install

Installation of OpenSSL library

DNP3 communication channels can be secured and the messages exchanged can be encrypted using TLS. OpenSSL is an open-source and mature library that implements various cryptographic functions and various utility functions. OpenSSL can be installed from the terminal using the following command:

```
sudo apt-get update
sudo apt-get install libssl-dev
```

The recommended version of OpenSSL for x86 architecture is **OpenSSL 1.0.1t**

The recommended version of OpenSSL for ARM architecture is **OpenSSL 1.1.0f**

Installation of Boost library

Boost is a set of libraries for C++ programming language that provides support for file processing and various other applications. Boost can be installed from the terminal using the following command:

```
sudo apt-get install libboost-dev
```

The recommended version of Boost for x86 architecture is **Boost 1.55**

The recommended version of Boost for ARM architecture is **Boost 1.62**

Installation of OpenDNP3 library

The DER Outstation uses the Automatak's OpenDNP3 library for the protocol implementation (https://www.automatak.com/opendnp3/). OpenDNP3 is a free, open-source, portable, scalable implementation of the DNP3 protocol stack. The OpennDNP3 library can be installed using following steps:

Step 1: Download the Opendnp3 library using the following command from the terminal:

```
git clone --recursive https://github.com/automatak/dnp3.git
```

The '--recursive' flag has to be placed after the URL on specific versions of 'git'

Step 2: Opendnp3 uses a build system generator called CMake. If cmake is not available, install using following command:

```
sudo apt-get install cmake
```

Step 3: Build the OpenDNP3 library using the following commands from the ~/dnp3/ directory

```
cmake ../dnp3 -DDNP3_DEMO=ON -DDNP3_TLS=ON
make
sudo make install
```

Installation of DER Outstation

The DER Outstation can be downloaded from EPRI's Github repository. All the dependencies mentioned earlier should be installed before building the Outstation. The source code can be downloaded from the following link:

https://github.com/epri-dev/der-dnp3-an2018

To build the Outstation, use the following command from the terminal

```
bash build.sh
```

3
RUNNING THE OUTSTATION

Navigate to the ~/der-an-2018 directory and run the following command from the terminal:

```
./der
```

If the compiler cannot find the shared libraries for the project, use the following command to load the library path and run the executable again

```
export LD_LIBRARY_PATH=”/usr/local/lib”
```

Exercise caution when the above command is copied and pasted to a terminal. The double quotes (”) in MS-Word uses a proprietary encoding and can be interpreted differently by Linux terminals.

Using Transport Layer Security

Once the outstation is launched, the user is prompted with the option of creating TLS channels. The user can opt to launch TLS channels or bypass it to create regular TCP channels. If the user opts to create TLS channels, the following files must be put in the **der-an-2018\TLS** directory

Table 3-1
TLS files

File Type	File Name	Description
Trusted Certificate	ca.crt	Certificate file used to verify the peer or server. Can be CA file or a self-signed cert provided by other party
Local Certificate	ia_master_chain.pem	File that contains the certificate (or certificate chain) that will be presented to the remote side of the connection
Private Key	ia.key	File that contains the private key corresponding to the local certificate

Below is the screen capture of the Outstation creating the TLS channels

```
pi@raspberry:~/Desktop/dnp3-an-2018/der-an-2018 $ ./der
DER App Note 2018 Outstation !!!!
Do you want to create a TLS channel?[Y/N]
ms(1546556102019) INFO    manager - Starting thread (0)
Y
Creating TLS Server
Using CA certificate: TLS/ca.crt
Using certificate chain: TLS/ia_master_chain.pem
Using private key file: TLS/ia.key
channel state change: OPENING
ms(1546556104037) INFO    server - Listening on: 0.0.0.0:20000
```

Importing Measurement Values

The Analog Inputs (AI) and Binary Inputs (BI) can be updated to the outstation from the CSV file. In a real application, the downstream application (e.g. manufacturer software) that implements this outstation shall update these measurement values. For testing purposes, these values are imported from a CSV file. Once the communication channels are launched, the user can choose to import the measurement values using the following options:

Table 3-2
Measurement Read Options

Option	Description
a	Update Analog Inputs to Outstation
b	Update Binary Inputs to Outstation
c	Update Analog Outputs to Outstation
d	Update Binary Outputs to Outstation

Below is the screen capture of the options to import measurement values

```
pi@raspberry:~/Desktop/dnp3-an-2018/der-an-2018 $ ./der
DER App Note 2018 Outstation !!!!
Do you want to create a TLS channel?[Y/N]
ms(1548859981962) INFO    manager - Starting thread (0)
N
Creating TCP Server
channel state change: OPENING
ms(1548859983547) INFO    server - Listening on: 0.0.0.0:20000
In DER AN2018 outstation. Select an option and then press <enter>
a = Update Analog Inputs to Outstation, b = Update Binary Inputs to Outstation
c = Update Analog Outputs to Outstation, d = Update Binary Outputs to Outstation
'quit' = exit
```

4

WRITING A DER OUTSTATION APPLICATION

It's essential for a developer to understand the DNP3 Outstation configuration, architecture and interactions to develop the DER application that implements the AN2018 DER Outstation.

Setting up the Communication Channel

The DNP3 Outstation stack is set up with a TCP/TLS server that will listen for requests from Master Station. The IP address and Port number for the server can be configured in *`DEROutstation.cpp`*. The default IP address and port number is *0.0.0.0* and *20000* respectively.

```
channel = manager.AddTCPServer("server", FILTERS,
ServerAcceptMode::CloseExisting, "0.0.0.0", 20000,
PrintingChannelListener::Create());
```

TLS Channel

Alternatively, the Outstation can be set up with a TLS server for secured communication as shown below

```
channel = manager.AddTLSServer("server", FILTERS,
ServerAcceptMode::CloseExisting, "0.0.0.0", 20000, TLSConfig(caCertificate,
certificateChain, privateKey,2), PrintingChannelListener::Create(),ec);
```

The certificates and keys described in the previous section for creating a TLS channel should be placed in the following project directory

`dnp3-an-2018/der-an-2018/TLS`

DER Outstation API

The DER Outstation application will implement the interface `IDERCommandHandlerCallback` to receive the packaged smart function commands from the Outstation. `IDERCommandHandlerCallback` interface supports the smart functions that have been mandated in IEEE 1547 Standard for DER Communication. The interface `IDERCommandHandlerCallback` contains the function declarations or signature and the downstream application implementing the interface must have the definition and processing logic for the functions specified in the interface.

The code snippet below shows the function signature of Active power limit profile in the `IDERCommandHandlerCallback` interface

```
virtual void EnableActiveLimitPower(double
activePowerLimitDisCharge, double activePowerLimitCharge) = 0;
```

A default DER Outstation application `DERCommandHandlerCallbackDefault` that implements this interface was developed for testing and logging purposes. The smart function `EnableActiveLimitPower` is shown as example below. The downstream Outstation application that implements the interface will be notified via this function when a master station dispatches and activates an active power limit profile. The downstream application should implement the logic to process the profile parameters for each function.

```
void DERCommandHandlerCallbackDefault:EnableActiveLimitPower(double
activePowerLimitDisCharge, double activePowerLimitCharge){
        std::cout << "Enable Active Power Limit function \n";
        // Downstream application logic to process profile commands
}
```

The Interface `IDERCommandHandlerCallback` can be extended to support other smart functions as required by the downstream application by creating new relevant function signatures using the existing ones as a template.

Updating Measurement Values Back to Outstation

The measurement and monitoring values from the downstream application can be updated to Outstation by making a function call to one of the following functions in *`DEROutstation`*

- `UpdateBinaryInputs(vector<BinaryValues> values)`
- `UpdateAnalogInputs(vector<AnalogValues> values)`
- `UpdateBinaryOutputs(vector<BinaryValues> values)`
- `UpdateAnalogOutputs(vector<AnalogValues> values)`

The *AnalogValues* and *BinaryValues* are data structures with following fields in them:

```
struct AnalogValues{
    u_int16_t index;
    u_int8_t quality;
    double val;
};
struct BinaryValues{
    u_int16_t index;
    u_int8_t quality;
    bool val;
};
```

Extending Outstation to Support Additional DNP Points

The Outstation currently supports Analog and Binary points that are mandated in IEEE 1547 standard for DER Communication. However, as required by the developer, it can be extended to support additional points from the Application Note. The `DERAppNoteDefinition` contains the data structures objects of the all the Analog and Binary points defined in the application note.

The following structure object shows a subset of Analog Outputs points that are defined in the `DERAppNoteDefinition.cpp`

```
const AnalogOutputPointDefinition AOPoints[MAX_AOPOINTS] =
{
      {0,"RefVolt",0,2147483647,true,29},
       {1,"RefVoltOffset",-2147483648,2147483647,true,30},
       {2,"NominalGridFreq",0,70000,false,65535},
       {3,"OpenLoopResponseTMPerc",0,1000,false,65535},
       {4,"PFSignconvention",1,2,false,65535},
       {5,"RefforReAPSetPs",0,3,false,65535},
 }
```

The fifth column in each row, which is a binary bit defines if a point is supported by the Outstation. In the list of points shown above, points 2 to 5 are not supported in the Outstation as this field is set to *`false`*. This field should be set to *true* for the outstation to handle and process the DNP point. Similarly, the other DNP point types (Analog Input, Binary Output/Input) structure objects can be modified to add support for additional points.

Curve Functionality

Many of the DER operating modes require the use of a curve, i.e. a set of “X-values” and “Y-values” expressing a relationship between a measured quantity and a desired output.

Outstation can store and handle up to 100 generic curves and the master can define up to 100 points for each curve.

The flow chart below shows the process flow on how the master would write, update and enable a generic curve in the Outstation.

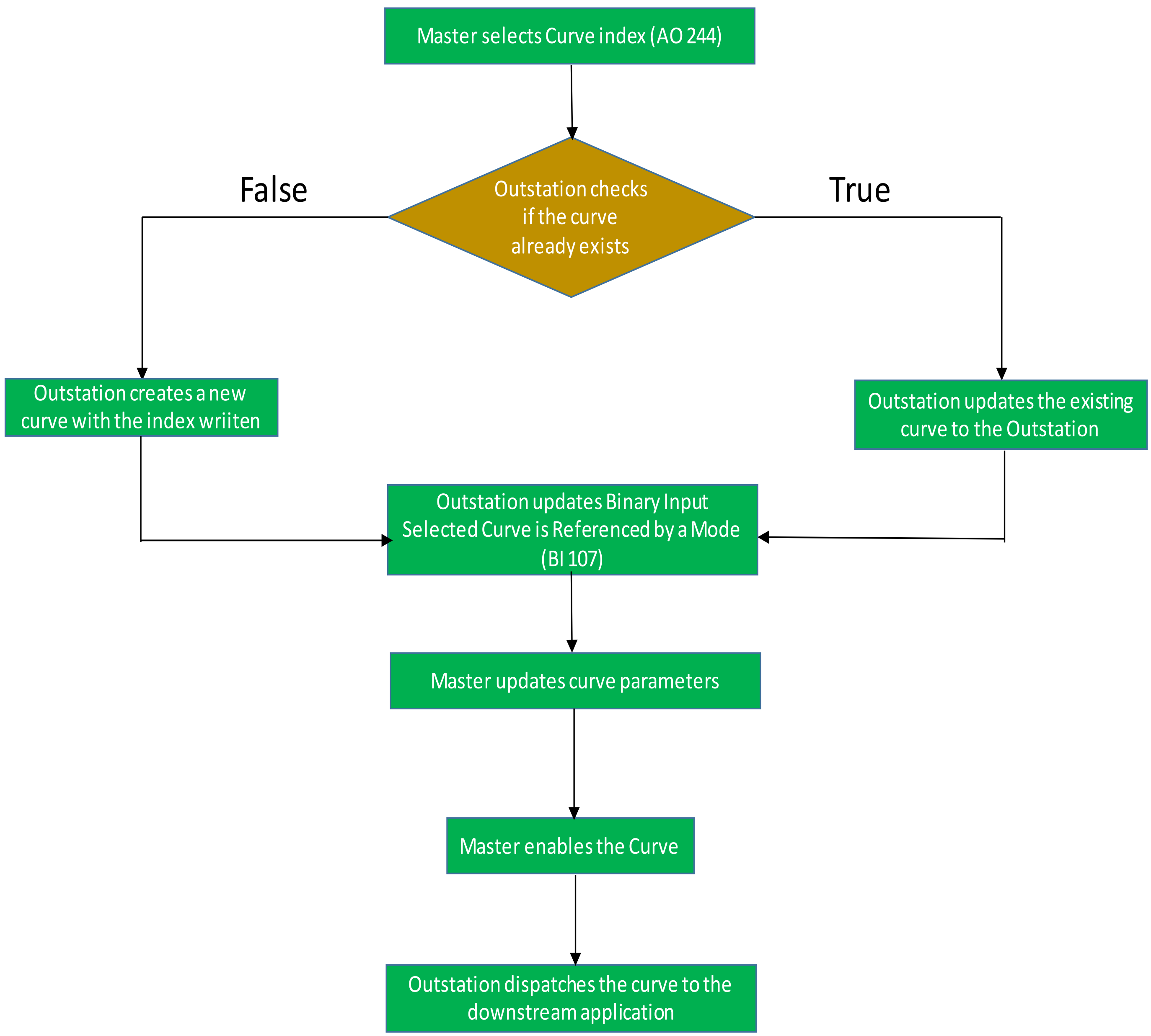

Figure 4-1
Flow Chart – Write, Update and Enable a Generic Curve in the Outstation

Schedule Functionality

A schedule is a curve in which the X-value is a Time Offset (specified as a number of seconds) from the start of the schedule and the Y-value (Schedule Value) is either:

- The set point for one of the core operating modes in the profile, or
- A value indicating a mode should be turned on or off using its existing configuration parameters; for instance, applying whichever curve index the master most recently set for that mode.

Outstation can store and handle up to 10 schedules and the master can define up to 100 points for each schedule.

The flow chart below shows the process flow on how the master would write, update and enable a schedule in the Outstation.

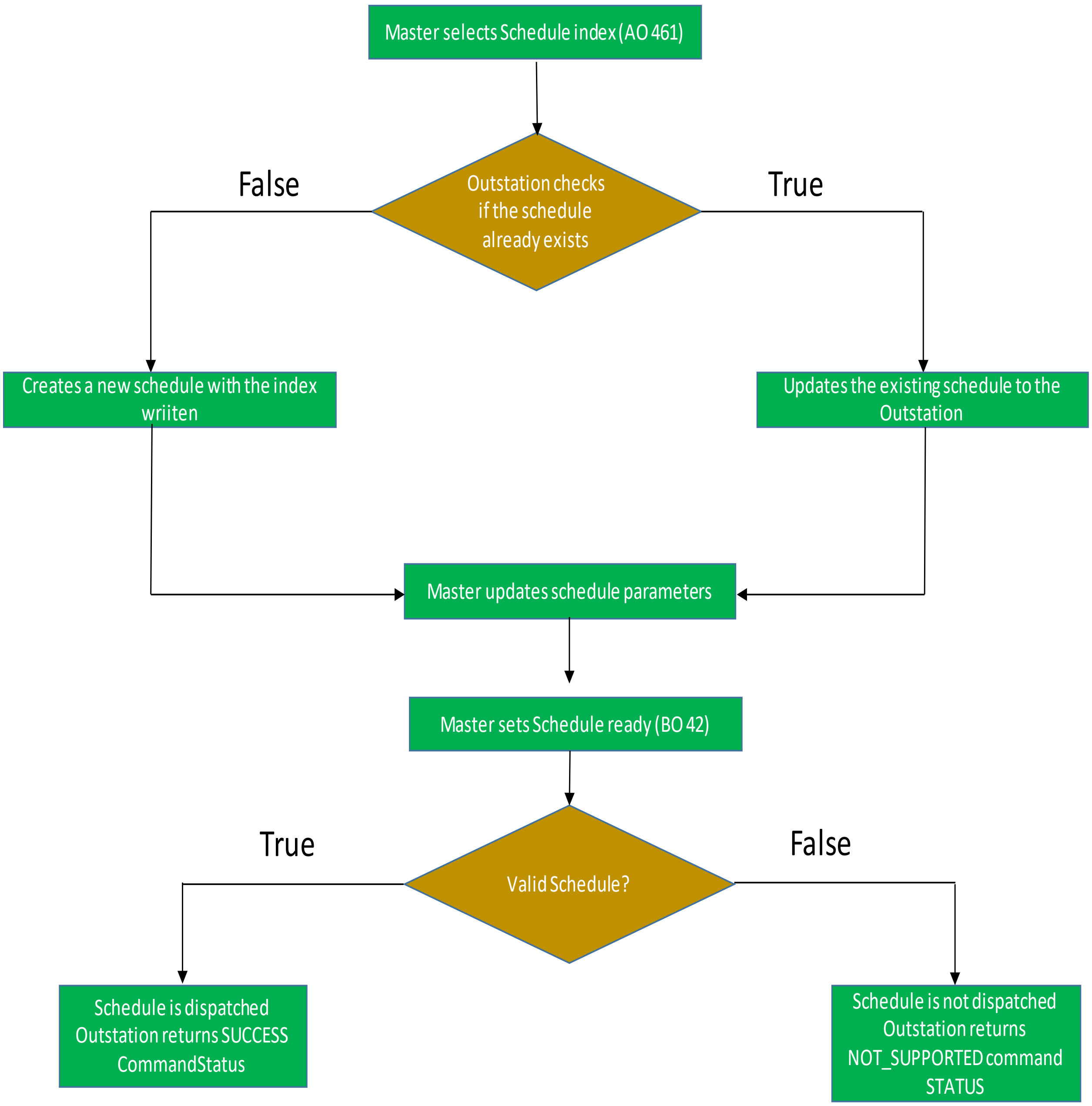

Figure 4-2
Flow Chart – Write, Update and Enable a Schedule in the Outstation

The below flow chart illustrates the validations the Outstation carries out on the selected schedule before it is dispatched to the downstream application.

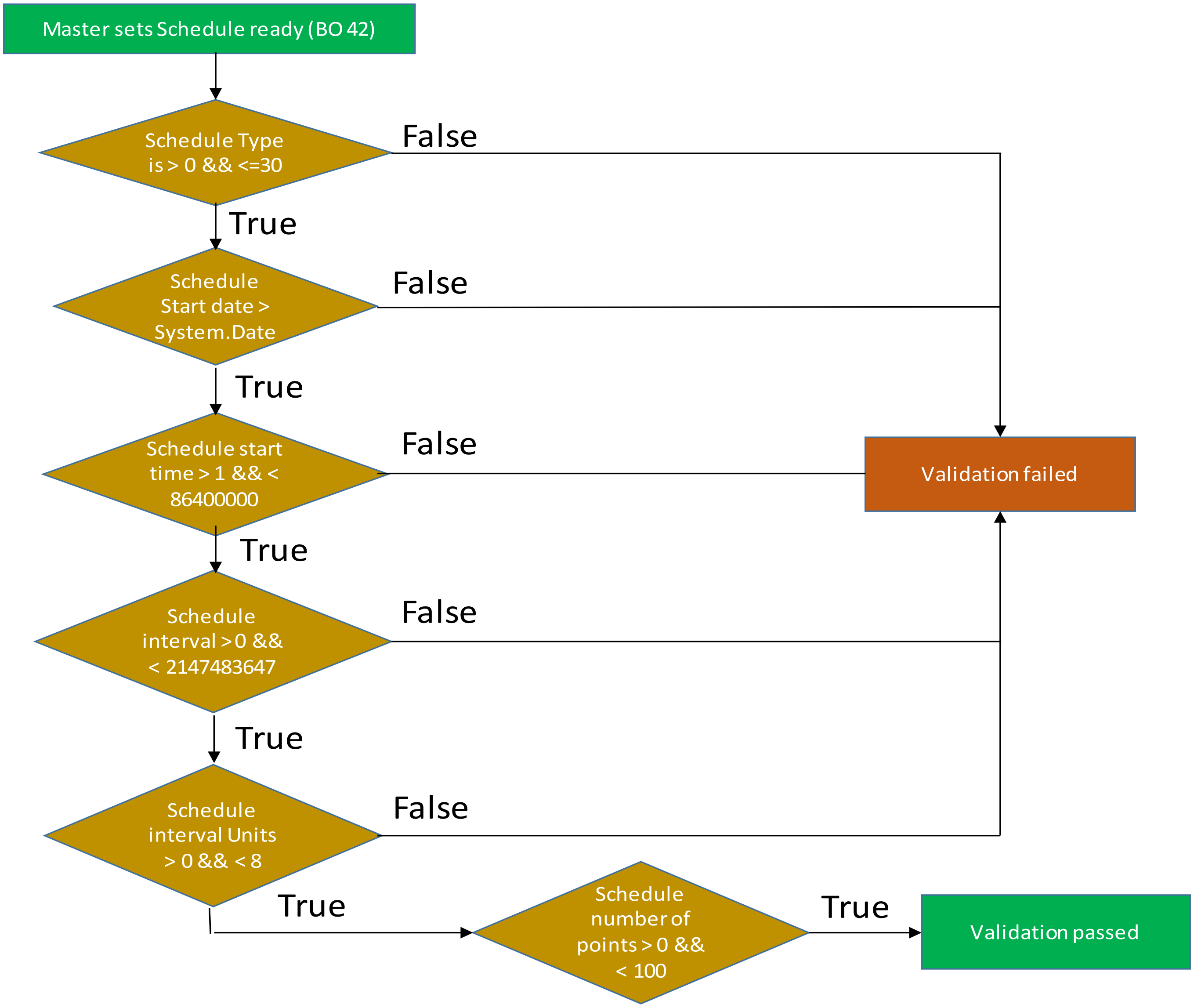

Figure 4-3
Flow Chart – Outstation's Validation of Schedule

The Electric Power Research Institute, Inc. (EPRI, www.epri.com) conducts research and development relating to the generation, delivery and use of electricity for the benefit of the public. An independent, nonprofit organization, EPRI brings together its scientists and engineers as well as experts from academia and industry to help address challenges in electricity, including reliability, efficiency, affordability, health, safety and the environment. EPRI members represent 90% of the electric utility revenue in the United States with international participation in 35 countries. EPRI's principal offices and laboratories are located in Palo Alto, Calif.; Charlotte, N.C.; Knoxville, Tenn.; and Lenox, Mass.

Together…Shaping the Future of Electricity

3002015355

Electric Power Research Institute
3420 Hillview Avenue, Palo Alto, California 94304-1338 • PO Box 10412, Palo Alto, California 94303-0813 • USA
800.313.3774 • 650.855.2121 • askepri@epri.com • www.epri.com