TEST SPECIFICATION

FIELDCOMM GROUP™

Connecting the World of Process Automation

HART®

COMMUNICATION PROTOCOL

HART EDD Test Specification

FCG TT20019
Edition 1.0
28 Jan 2020
RELEASED

Document Distribution / Maintenance Control / Document Approval

To obtain information concerning document distribution control, maintenance control, and document approval please contact FieldComm Group at the address shown below.

Copyright © 2020 FieldComm Group

Trademark Information

FieldComm Group™, FOUNDATION™ Fieldbus and HART-IP™ are trademarks, and HART®, *Wireless*HART®, ROM® and SIF® are registered trademarks of FieldComm Group, Austin, Texas, USA. Any use of these terms hereafter in this document, or in any document referenced by this document, implies the trademark/registered trademark. All other trademarks used in this or referenced documents are trademarks of their respective companies. For more information, contact FieldComm Group at the address below.

FieldComm Group
Attention: President and CEO
9430 Research Boulevard
Suite 1-120
Austin, TX 78759, USA
Voice: (512) 792-2300
FAX: (512) 792-2310

http://www.fieldcommgroup.org

Intellectual Property Rights

The FieldComm Group (the Group) does not knowingly use or incorporate any information or data into the HART, FOUNDATION Fieldbus and FDI protocol standards, which the Group does not own or have lawful rights to use. Should the Group receive any notification regarding the existence of any conflicting private IPR, the Group will review the disclosure and either (A) determine there is no conflict; (B) resolve the conflict with the IPR owner; or (C) modify the standard to remove the conflicting requirement. In no case does the Group encourage implementers to infringe on any individual's or organization's IPR.

TABLE OF CONTENTS

1 Scope

This document defines the test criteria for FieldComm Groups' HART EDD technology. Tests are comprised of manual source code inspection and automated verifications performed by the latest released EDD developer tools.

2 Normative References

Specifications	**Document Number**
HART Communication Protocol Specification	FCG TS20013
Unified EDDL	FCG TS61804-3, -4, -5
HART EDDL Specifications	FCG TS20500, FCG TS20501
HART Common Tables Specification	FCG TS20183
HART EDD Registration Policy	FCG PD20018
Command Summary Specification	FCG TS20099
Slave Universal Command Test Specification	FCG TT20005

3 Definitions

Burst Mode A special mode of a slave device which repeatedly sends the response to a selected HART command without the need for a request from a master.

Burst Mode Device is a slave device that repeatedly issues the reply to a selected command. Once a slave is in this (optional) mode, the reply is sent without any further action by any master. A bursting slave is like a tertiary master because it also initiates channel activity.

Device is a HART compatible Slave or Master.

Device Revision The integer returned in byte 5 of Identity Commands (see the Command Summary Specification). This defines the revision level of the command set supported by the field device including the device-specific commands. The Command Summary Specification defines when a new Device Type number must be allocated (as opposed to the Device Revision being incremented).

Device Type The integer returned in bytes 1-2 of Identity Commands (see the Command Summary Specification). This defines the command set supported by a device. The Command Summary Specification defines when a new Device Type number must be assigned (as opposed to the Device Revision being incremented). Device Type codes are controlled and must be allocated by the FieldComm Group.

Device Variable A uniquely defined data item within a Field Device that is always associated with cyclical process information. A Device Variable's value varies in response to changes and variations in the process. All HART compatible field devices contain Device Variables. However, simple field devices may use only Dynamic Variables and not expose the underlying Device Variables at the Protocol Application Layer interface.

Dynamic Variable The connection between the process and an analog channel. All HART field devices may contain Primary, Secondary, Tertiary, and Quaternary Variables that are mapped to the first 4 analog channels in a field device. These are collectively called the Dynamic Variables. The Primary Variable is always supported and is connected to the first 4-20mA channel, the same channel that always supports HART communication. The SV, TV, and QV may or may not be supported and, furthermore, may not have an associated Analog Channel.

EDDL File A source code of the EDD in extension of either .ddl (only one file) or .dd (possibly none, one, or many).

Enumeration A numerical list where each number corresponds to a specific function or text string. Basically, an enumeration is a look-up table (e.g., of engineering unit codes) with each row containing a number and the corresponding function of text string.

Failure Point is a unique identifier for a test point.

Manufacturer ID The integer returned in byte 17-18 of Identity Commands. See the Command Summary Specification for details on Identity Commands. Manufacturer IDs are controlled and must be allocated by the FieldComm Group.

May or **Optional** identifies a requirement that is completely optional and can be supported at the discretion of the implementation.

Must, Shall, or **Required** denotes an absolute mandatory requirement.

Should or **Recommended** indicates a requirement that, given good cause/reason, can be ignored. However, the consequences of ignoring the requirement must be fully understood and well justified before doing so.

Test Point is the description of the inspection performed on the EDD source or supporting files.

4 Abbreviations

EDD	Electronic Device Description
EDDL	Electronic Device Description Language
FCG	FieldComm Group
TFP	Test Failure Point

5 Basic Elements of Device Description

5.1 Identification

Each Device Description is identified using the manufacturer line that specifies the manufacturer, device type, device revision and the DD revision of the DD.

The following test steps are applied to the manufacturer line.

5.1.1 Test ID: TC_511- Syntax of MANUFACTURER Declaration

This test verifies if DD uses the standard Syntax for MANUFATURER line.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	6.1
HART EDD Registration Policy	1.0	

Standard Syntax:

MANUFACTURER *manufacturer-string*, **DEVICE_TYPE** *device-string*,
DEVICE_REVISION *X (decimal)*, **DD_REVISION** *X (decimal)*

Where devices.cfg defines:

MANUFACTURER 0x00**yyyy** = *manufacturer-string*
DEVICE_TYPE 0x**zzzz** = *device-string*

Where **yyyy** and **zzzz** are both defined in hexadecimal characters.

Prerequisites/Setup

- Open EDD source files.

Test Type: Automatic and Manual

Test objective: Syntax of manufacturer line must follow standard syntax defined. Manufacturer line is case sensitive. Manufacturer name and device type are format sensitive.

Test Procedure:

- Verify manufacturer-string is defined in the device.cfg file so that “ERROR 531: unknown manufacturer xxxxx” and “ERROR 530: file-name, line line-number: Syntax error” is not displayed *(TFP511_001)*
- Verify device-string is defined in the device.cfg file so that “ERROR 531: unknown manufacturer xxxxx” and “ERROR 530: file-name, line line-number: Syntax error” is not displayed *(TFP511_002)*
- Verify the first character of the MANUFACTURER line is not a “ “ space character. *(TFP511_004)*
- Verify DEVICE_TYPE name exactly matches previously submitted devices.cfg DEVICE_TYPE name. Note: If the DEVICE_TYPE entry in the devices.cfg file is updated, this affects all device revisions. Future submissions must be updated to maintain the same devices.cfg entry. *(TFP511_005)*
- Verify device-string begins with an underscore “_” character. Underscore “_” is necessary for specific FieldComm Group repository tools. *(TFP511_006)*

- Verify DD_REVISION number incremented for a DD Revision registration since the last registration (Note: This is to avoid duplicate encoded DD file names) and DD Revision registration does not increment DEVICE_REVISION. *(TFP511_007)*
- Verify DEVICE_REVISION number incremented for a Device Revision registration since the last registration (Note: This is to avoid duplicate encoded DD file names) *(TFP511_008)*
- Verify legal trademarked name “HART” does not appear in the manufacturer line. *(TFP511_009)*

5.1.2 Test ID: TC_512- Duplication of MANUFACTURER construct.

Purpose of this test is to avoid repetition of MANUFACTURER line in the DD.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files.
- Tokenize DD using latest version of released tokenizer

Test Type: Manual

Test objective: Verify the DD implemented a unique manufacturer and device identification.

Test Procedure:

- Verify DD uses a unique and single MANUFACTURER line *(TFP512_001)*
- Verify preprocessor “__TOKVER__ “ is not used to include multiple MANUFACTURER lines *(TFP512_002)*
- Verify conditional statements is not used on the MANUFACTURER line *(TFP512_003)*

5.2 Import

Standard DDs model the HART application layer. IMPORT simplifies and reduces the amount of original development and minimizes errors required to produce a new or revised Device Description. All imports defined below are mandatory, however Common Practice may be excluded if there are no Common Practice Commands implemented in the EDDL.

5.2.1 Test ID: TC_521-Test Mandatory Standard IMPORTs

This test verifies code development includes the mandated Standard DD IMPORTs.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual

Test Objective: Verify the four mandatory IMPORTs. "_TABLE", "_UNIVERSAL" and "_PV" are included. Verify "_COMMON_PRACTICE" is implemented if the device and EDDL contains any Common Practice Commands.

Test Procedure:

- Verify _TABLE IMPORT construct is defined in the DD *(TFP521_001)*
- Verify _UNIVERSAL IMPORT construct is defined in the DD *(TFP521_002)*
- Verify _PV IMPORT construct is defined in the DD *(TFP521_003)*

Verify _COMMON_PRACTICE IMPORT construct is present in the DD. Applicable only if the device supports Common Practice Commands and the commands are included in the EDDL. *(TFP521_004)*

5.2.2 Test ID: TC_522- Duplicated IMPORTS

Imports of the Standard DDs is mandatory but repetition is not permitted. Verify a single IMPORT is used for each required Standard Import. Each import as listed in the *HART EDD Registration Policy* is permitted only once in a DD. EDD must be compactable with a unique HART version. There are sets of standard DDs that must be used together and must not be mixed.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	(Table 2)

Prerequisites/Setup

- Open EDD source files..

Test Type: Manual and Automatic.

Test Objective: Verify the proper implementation of Standard Library IMPORTs, and verify they are not duplicated.

Test Procedure:

- Verify IMPORT construct with each of the four import-strings (_UNIVERSAL, _PV, _TABLES, _COMMON_PRACTICE) are used only once in a DD. *(TFP522_001)*
- Verify preprocessor conditional statements is not used around the IMPORT constructs (i.e. To make DD compatible with multiple HART versions, conditional statements in the program that switches between HART 7 or HART 5 statement are not permitted) *(TFP522_002)*
- Verify that IMPORT of a non-standard DD is restricted to a previous DEV_REV of the same DEV_TYPE *(TFP522_003)*
- Verify that proper combinations of Standard DD file versions are used (Note: See HART EDD Registration Policy Table 2.) *(TFP522_004)*
- Verify preprocessor "__TOKVER__" is not defined around the Standard IMPORT constructs *(TFP522_005)*

6 Data Modeling

Data modeling is central to any DD and all data in the device should be modelled in its DD. Test cases included in this section verifies organization of the data in the device and relations used to model interdependencies between data.

6.1 Variable

The variable construct is the principle mechanism for modeling data items found in the devices real time database. Every variable must have a name and may be used elsewhere in DD to refer to variables. There are fourteen (14) attributes supported for variables: LABEL, HELP, CLASS, VALIDITY, VISIBILITY, PRIVATE, HEIGHT, WIDTH, HANDLING, DEFAULT_VALUE, TYPE, pre or post actions, REFRESH_ACTIONS and CONSTANT_UNIT. All variables must have TYPE and all other attributes are optional.

The following test cases are applied to common attribute line.

6.1.1 Test ID: TC_611- TYPE Attribute in VARIABLE Constructs

This test case verifies mandatory attribute "TYPE" and the syntax.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	8.1
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files..

Test Type: Manual and Automatic.

Test Objective: Verify "TYPE" is included as an attribute for VARIABLE construct.

Test Procedure:

- Verify attribute "TYPE" is defined in all non-Standard VARIABLE constructs. *(TFP611_001)*
- Verify "REDEFINE TYPE" is not used for an imported Standard VARIABLE. The REDEFINE, if required, must be on the contents of the item.. *(TFP611_001)*
- Verify MIN_VALUE is less than MAX_VALUE *(TFP611_002)*
- Verify Variables use a valid variable TYPE *(TFP611_003)*
- Verify the number of MIN_VALUEs defined for a VARIABLE is not greater than the number of MAX_VALUEs defined for that VARIABLE. (i.e. they need to be evenly paired.) so that "ERROR 623: *Invalid number of MIN/MAX values defined for VARIABLE variable-name"* is not displayed. *(TFP611_004)*
- Verify that there are not duplicate MIN_VALUEx or MAX_VALUEx so that "ERROR 635: VARIABLE variable-name has duplicate definition of min/max value which-value" is not displayed. *(TFP611_005)*
- Verify the variable TYPE does not contain pre-processor conditions (#if/#endif). *(TFP611_006)*

6.1.2 Test ID: TC_612- CLASS Attribute in DD VARIABLE Constructs

Use of the CLASS attribute must be used as defined in the specification.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	A.1
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files..

Test Type: Automatic.

Test Objective: Verify proper use of the CLASS attribute in DD the construct.

Test Procedure:

- Verify that CLASS LOCAL (LOCAL_A) is not combined with other Classes so that "ERROR 336: CLASS LOCAL cannot be mixed with other CLASSes" is not displayed. *(TFP612_001)*
- Verify that CLASS does not have multiple functional classes defined so that "ERROR 335: Multiple Functional Class values are not allowed" is not displayed. *(TFP612_002)*
- Verify that conditional CLASSes (via preprocessors directives) are not used *(TFP612_003)*

6.1.3 Test ID: TC_613- HANDLING Attribute in DD VARIABLE Constructs

Use of the "HANDLING" attribute must be as defined in the specifications.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	

Prerequisites/Setup

- Open EDD source files..

Test Type: Automatic and Manual

Test Objective: Verify proper use of the HANDLING attribute in the DD VARIABLE construct imported from a Standard DD.

Test Procedure:

- Verify a Standard Variable HANDLING attribute is not REDEFINEd from "READ" to "WRITE "when only a READ COMMAND exists for this Variable. *(TFP613_001)*
- Verify a Standard Variable HANDLING attribute is not REDEFINEd from "READ" to "READ & WRITE" without a WRITE COMMAND defined for this Variable. Not applicable to CLASS LOCAL Variables. *(TFP613_002)*
- Verify a Standard Variable HANDLING attribute is not REDEFINEd from "WRITE" to "READ" while a Write Command is defined for the Variable. Not applicable to CLASS LOCAL Variables. *(TFP613_003)*

6.1.4 Test ID: TC_614- LABEL Attribute in DD VARIABLE Constructs

Use of the "LABEL" attribute must be as defined in the specifications.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files..

Test Type: Manual

Test Objective: Verify proper use of the LABEL attribute in the DD variable construct.

Test Procedure:

- Verify conditional statements (IF/ELSE, SELECT/CASE) are not used for the LABEL attribute. *(TFP614_001)*
- Verify the imported Variable LABEL is not redefined.(Note: Command 48 *device_specific_status_x* variables are expected to be redefined and are not applicable here.) *(TFP614_002)*

6.1.5 Test ID: TC_615- VALIDITY VISIBILITY Attribute in DD ITEM Constructs

Use of the "VALIDITY" attribute must be as defined in the specifications.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files..

Test Type: Manual

Test Objective: Verify proper use of the VALIDITY/ VISIBILITY attribute in the DD item construct.

Test Procedure:

- Verify where VALIDITY/VISIBILITY is defined, both TRUE and FALSE are not simultaneous possible results. *(TFP615_001)*
- Verify VALIDITY/VISIBILITY is not added to a mandatory root menu. *(TFP615_002)*
- Verify VALIDITY/VISIBILITY is not added to an imported Standard DD VARIABLE. (Note: All Standard Variables must always *exist* in the EDD. Not all Standard Variables need to be *VISIBLE* or *REFERENCED*.) *(TFP615_003)*

6.2 Command

6.2.1 Test ID: TC_621-Standard Command Modeling Rules

This test case verifies standard Command structure, range of Commands, and Command transactions.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	
HART Command Summary Specification	10.0	7.1

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual and Automatic.

Test Objective: Use of an IMPORTED “COMMAND” must be compliant with the Standard Command defined in the Command specifications. Test does not consider device-specific commands.

Test Procedure:

- Verify a command number is a not a duplicate of another Command so that “ERROR 663: COMMAND command-name NUMBER number-number duplicates COMMAND command-name NUMBER” Is not displayed *(TFP621_001)*
- Verify a Command name is consistent between import and construct (i.e., Command name must be defined in the Standard DD that is being imported). *(TFP621_002)*
- Verify a Command number is consistent between import and construct (i.e., Command number must be defined in the Standard DD that is being imported). *(TFP621_003)*
- Verify an Imported Command is not REDEFINED to contain user defined VARIABLEs and/or constants (i.e., no replacements nor additions are permitted for Standard Commands.) *(TFP621_004)*
- Verify a Command number which is not imported is within the allowed NUMBER range defined by the “HART Command Number Partitions” *(TFP621_005)*
- Verify Universal and Common Practice Commands were imported (i.e., Copy/paste directly into the DD source is not permitted.) *(TFP621_006)*
- Verify a Command Transaction which is imported and identified as truncatable must use all or a subset (may not re-order the data items) of the VARIABLEs (and its qualifiers) referenced in the Standard DD only occurs if it is identified as Truncatable *(TFP621_007)*
- Verify an imported Command with Transaction(s) identified as non-truncatable is not redefined. Truncatable Commands are numbers 3,8,9, and 48. *(TFP621_008)*
- Verify a WRITE Command contains the same data items (and in the same order) in the Request as those in the Response Data. HART Protocol requires autonomous commands. *(TFP621_009)*
- Verify there are no duplicate Command attributes (i.e., REQUEST, REPLY, or RESPONSE_CODES) in the same Command Transaction so that *“ERROR 324: transaction-sub attribute is already defined for transaction”* is not displayed *(TFP621_010)*
- Verify no value is used for multiple response codes to specify a DD item so that “ERROR 606: Multiple response codes in item-name have the same value integer” is not displayed. *(TFP621_011)*
- Verify there are no duplicated TRANSACTION keywords (Transaction number) for the Command so that “ERROR 643: COMMAND command-name TRANSACTION transaction-number is multiply defined” is not displayed. *(TFP621_012)*
- Verify all entities referenced in the Command Transactions are defined in the current Device Description or an imported Device Description so that *“ERROR*

645: COMMAND command-name ID command-id has no results possible" is not displayed. (*TFP621_013)*

- Verify no Response Code description field is defined with an empty string so that "ERROR 646: construct-identifier response code response-code-value description not specified" is not displayed. *(TFP621_014)*
- Verify all Command Transactions include a REPLY, and a null REPLY was not used (REPLY {}) so that "Error 648: COMMAND command-name TRANSACTION transaction-number must have a REPLY" is not displayed. *(TFP621_015)*

6.2.2 Test ID: TC_622- General Data Modeling Construct Rules

Here are commonly used data modeling constructs which were not already defined above.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files.

Test Type: Automatic

Test Objective: To verify general data objects follow the DD construct modeling rules.

Test Procedure:

- Verify there are no duplicated attribute definitions so that "*ERROR 312: file-name, line line-number: attribute of item-name is already defined*" is not displayed. Same attribute cannot be included more than once for a single item. *(TFP622_001)*
- Verify other output status classes (DVn,TVn or AOn) are not used if "ALL" is already used so that *"ERROR 326: ALL not allowed with output-status"* is not displayed. *(TFP622_002)*
- Verify no deprecated HART attributes were used so that *"ERROR 334: Invalid attribute reference".* is not displayed, this includes READ_TIMEOUT and WRITE_TIMEOUT which are no longer supported. *(TFP622_003)*
- Verify DD Items were properly separated with a comma "," so that *"ERROR 342: Comma is missing in the list* "is not displayed. *(TFP622_004)*

  Verify no attribute keywords were duplicated (Example: *HANDLING READ & WRITE & READ*, and *CLASS INPUT & DIAGNOSTIC & SERVICE & INPUT*.) so that *"ERROR 308: file-name, line line-number: stack-overflow appears more than once or file-name, line line-number: syntax-error appears more than once*" is not displayed. *(TFP622_005)*
- Verify variable size is non-zero so that "ERROR 310: file-name, line line-number: VARIABLE variable-name cannot have size zero" is not displayed. *(TFP622_006)*
- Verify an item (i.e., VARIABLE, MENU, etc.) is not referenced in the LABEL attribute. *(TFP622_007)*
- ELEMENTS with same index cannot be specified more than once in the same ARRAY. Verify there are no duplicated ARRAY index numbers so that

"ERROR 605: Multiple ELEMENTs of ARRAY item-name have the index integer" is not displayed. *(TFP622_008)*

- Verify there are no duplicated enumerations within a single Variable so that *"ERROR 607: Multiple enumerations of VARIABLE variable-name have the value integer*" is not displayed. *(TFP622_009)*
- Verify that a "Method" CLASS is not defined as an "Application CLASS" (DIAGNOSTIC, DYNAMIC, SERVICE, IS_CONFIG, SPECIALIST). (*TFP622_010)*
- Verify that a "Method" CLASS is not defined as LOCAL. *(TFP622_011)*
- Verify that a "Method" CLASS is not INPUT nor OUTPUT. (*TFP622_012)*
- Verify Variable does not have both a CONSTANT_UNIT and UNIT relation defined, so that "*ERROR 616: Variable variable-name has both CONSTANT_UNIT and a UNIT relation defined*" is not displayed. *(TFP622_013)*
- Verify that an attribute for an item which cannot be conditional is not conditional so that "*ERROR 626: attribute for item-name is conditional*" is not displayed. *(TFP622_014)*
- The CONSTANT_UNIT attribute must be a constant. Verify CONSTANT_UNIT is not used as a conditional value So that *"ERROR 673: Non-constant CONSTANT_UNIT attribute"* is not displayed. (*TFP622_015)*
- Verify the BLOCK or RECORD item is not used in the HART EDD. Remove the item or use #ifdef/#endif pre-processor block so that "*ERROR 700: file-name, line line-number: Found (BLOCK or RECORD) <name> in the HART device description"* is not displayed. (*TFP622_016)*
- Verify that a TIME_VALUE Variable with DISPLAY_FORMAT attribute defined must also define TIME_SCALE so that *"ERROR 705: Variable <name>: DISPLAY_FORMAT may not be present when TIME_SCALE is undefined"* is not displayed *(TFP622_017)*
- Verify that a TIME_VALUE Variable with EDIT_FORMAT attribute defined must also define TIME_SCALE so that "*ERROR 706: Variable <name>: EDIT_FORMAT may not be present when TIME_SCALE is undefined*" is not displayed. *(TFP622_018)*
- Verify that a TIME_VALUE Variable must not define both a TIME_FORMAT attribute and a TIME_SCALE attribute so that "*ERROR 708: Variable <name>: TIME_FORMAT may not be present when TIME_SCALE is supplied*" is not displayed *(TFP622_019)*
- Verify that a Unit Relation dominant Variable (i.e., cause-parameter, left of colon":") must not define TYPE non-ENUMERATED. *(TFP622_020)*

7 Use of References

7.1 Rules to Import the Standard DD Library

The IMPORT construct allows the reuse of standard source code. The HART Standard DD Library exists to eliminate the need for individual developers to define Commands, Variables, and Enumerations from the HART Protocol Specifications. The Standard DD Library encourages developers to use predefined program modules and enables modularity and re-usability of the program.

7.1.1 Test ID: TC_711-General IMPORT Construct Rules

This test case tests the source against the rules for Standard DD Library imports.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	
HART EDD Registration Policy	1.0	
HART Standard DD Library	Current	

Prerequisites/Setup

- Open EDD source files.

Test Type: Automatic

Test Objective: Verify the source code follows defined guidelines for successful and error free IMPORT.

Test Procedure:

- Verify the IMPORT directive "EVERYTHING" is not combined with another directive. (*TFP711_001)*
- Verify the latest released version of the Common Tables Specification Standard DD, Common Practice Specification Standard DD, PV Specification Standard DD, Universal Specification Standard DD are Imported. (*TFP711_002)*
- Verify imported VARIABLES are not deleted while they are actually used. (*TFP711_003)*
- Verify imported items belong to a Standard DD. (*TFP711_004)*
- Verify HART truncatable (or "redefined") Commands do not contain pre-processor directives (#if/#ifdef/#else). (*TFP711_005)*
- Verify IMPORT does not result in the following errors: "*ERROR 404: Could not find member member-name of item item-name in device device-id"* or *"ERROR 400: Import error in file-name, line line-number: could not find item item-name in device device-id to import*". (TFP*711_006*)
- Verify DDL items are defined or IMPORTed so that "*ERROR 509: Item item-name is used but not defined*" is not displayed. (*TFP711_007)*
- Verify there are no multiple redefinitions of same attribute for a specified item so that "*ERROR 609: Attribute of item-name redefined multiple times* **.**" is not displayed. (*TFP711_008)*
- Verify Universal or Common Practice Command(s) are not defined without using an IMPORT of the Standard DDs. "*ERROR 641: MANUFACTURER specified for device-specific command-type device-specific- command-number COMMAND command-name, not permitted*". (TFP*711_009*)

7.1.2 Test ID: TC_712-TABLE Import Construct Rules

This test case verifies TABLE import construct rules.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	
Common Table Specification	Current	

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual

Test Objective: Verify the source code follows guidelines for successful and error free TABLES DD import.

Test Procedure:

- Verify the IMPORT of TABLES DD uses the import directive "EVERYTHING" or macro ALL_TABLE or it uses one or more of the following macros: HART_DEVICE_TABLES, BURST_TABLES, WIRELESS_TABLES, IO_ADAPTER_TABLES and/or VAR_TREND_TABLES (*TFP712_001)*
- Verify all Variables imported via "Standard TABLES DD" are defined in the Tables import section. (*TFP712_003)*
- Verify the Device Type "NAME" used in VARIABLE "device_type" exactly matches the current *Common Tables Specification*. *(TFP712_004)*
- Verify the Company String Name used in VARIABLE "manufacturer_id" and "private_label_distributor" exactly matches the current *Common Tables Specification*. *(TFP712_00*5)

7.1.3 Test ID: TC_713-UNIVERSAL Import Construct Rules

This test case verifies UNIVERSAL import construct rules.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	(Table 3)
Universal Command Specification	Current	
Slave Universal Command Test Specification	4.1	7.13

Prerequisites/Setup

- Open EDD source files.
- Open HART Test System device test logs

Test Type: Manual

Test Objective: Verify the source code follows guidelines for successful and error free UNIVERSAL DD import.

Test Procedure:

- Verify the IMPORT of the UNIVERSAL DD includes “EVERYTHING”. *(TFP713_001)*
- Verify Command 48 has a minimum of 11 response bytes (see Table 1).
- Verify unsupported Command 48 device_specific_status_x bytes are deleted and the total number of bytes in the response matches the HART Test System device test logs (refer to UAL048A). *(TFP713_002)*
- Verify Command 48 byte descriptions do not contain conditional pre-processors (#if, #ifdef). *(TFP713_003)*
- Verify for the following mandatory Command 48 bytes: Verify the Variable attribute TYPE is not REDEFINEd. Verify these Variables are not DELETEd. *(TFP713_004)*

Table 1. Command 48 Response Bytes

All Devices Must Support	Device with >1 Analog
response_code, device_status, data Bytes 0-8	response_code, device_status, data Bytes 0-8
extended_field_device_status	extended_field_device_status
standardized_status_0	standardized_status_0
	standardized_status_1[1]
	standardized_status_2[1]
	standardized_status_3[1]
	analog_channel_saturated1[1]

- Verify Command 48 VARIABLE TYPE is not redefined from BIT_ENUMERATED to ENUMERATED, and the size is maintained as one (1). *(TFP713_005)*
- Verify Command 9 is not truncated to less than four (4) slot variables. *(Note: Four (4) device variable slots is mandatory even if the device support less than 4 device variables.)* Verify the maximum number of device variables represented in the DD for Command 9 matches the HART Test System device test logs for Byte 13 (maximum number of device variables) of Command 0 (refer to UAL000). *(TFP713_006)*

Table 2. Command 9 Device Variable Slots

4 Device Variables	Command 9 must support 4 Device Variable slots
5 Device Variables	Command 9 must support 5 Device Variable slots
6 Device Variables	Command 9 must support 6 Device Variable slots
7 Device Variables	Command 9 must support 7 Device Variable slots
8 Device Variables	Command 9 must support 8 Device Variable slots

---

[1] Must not replace Command 48 response bits with 0x00 if the bit is not supported by the device. The constant value 0x00 is only acceptable as a replacement for the “Operating mode” variable because this variable is no longer supported in Command 48.

>=9 Device Variables	Command 9 must support 8 Device Variable slots

- Verify Variables imported via “Standard UNIVERSAL DD” are not defined in other import sections. *(TFP713_007)*

- Verify that the Command 8 Command response references a constant data item[1] in the DD for any Dynamic Variable that is not supported by the device (refer to UAL012). *(Note: if the DD Developer does not do this, they will need to create a dummy collection for all Dynamic Variables which are truncated in Command 3)* *(TFP713_008)*

7.1.4 Test ID: TC_714-PV Import Construct Rules

This test case verifies PV import construct rules.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	

HART 5 does not support VARIABLE loop_flags so this may be DELETEd for HART 5 DDs. Furthermore, for this instance, the MEMBER ANALOG_CHANNEL_FLAGS for COLLECTION OF VARIABLE analog_io must also be DELETEd (see sample DD, sample1_r1.ddl).
UNIT and WRITE_AS_ONE relations may be DELETEd from the Standard _PV import, however this is not recommended.

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual

Test Objective: Verify the source code follows guidelines for successful and error free PV DD import.

Test Procedure:

- Verify the Verify the IMPORT of the Standard PV DD uses the keyword “EVERYTHING”. (*TFP714_001)*
- Verify that Device Variable Codes 244-250 are not implemented in the COLLECTION OF COLLECTION primary_variable, secondary_variable, tertiary_variable, quaternary_variable *(Note: For these collections, these codes affect Command 9 and could cause circular dependencies. Thus, these codes may be used elsewhere in the DD to support configuration of the device such as burst mode.)* (*TFP714_002)*
- For devices supporting Command 50 and 51, verify primary_variable_code, secondary_variable_code, tertiary_variable_code or quaternary_variable_code are used and not constant values (i.e. 0,1,2,3) (Referred to as Device Variable mapping). (*TFP714_003)*
- Dynamic Variable collection (secondary, tertiary or quaternary) is no longer referenced in the ARRAY OF COLLECTION dynamic_variables, and was not DELETEd. If supporting less than 4 Dynamic Variables, the developer must

---

[1] Dynamic Variables not supported by the device need to return **0 "Not Yet Classified"** for Command 8.

DELETE the unsupported collection for each unsupported Dynamic Variable Collection. (*TFP714_004)*

- COLLECTION OF VARIABLE “analog_io” and “scaling” used the REDEFINE directive to ADD additional MEMBERs, while Command 60 series of Commands were not imported. If two or more loop currents supported, then the developer must ADD the additional loop current(s) to the PV standard. (*TFP714_005)*

7.1.5 Test ID: TC_715-Common_Practice Import Construct Rules

This test case verifies Common Practice Command import construct rules.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual

Test Objective: Verify the source code follows guidelines for successful and error free COMMON_PRACTICE DD import.

Test Procedure:

- Common Practice Commands were defined outside of the COMMON_PRACTICE IMPORT section. (*TFP715_001)*

8 Redefinition Rules

Any alteration or re-definition must meet all requirement found in the latest release of the HART Protocol Specifications.

8.1 Redefinition Rules for a DD Item

There are rules to be followed to redefine DD items.

8.1.1 Test ID: TC_811- Standard VARIABLE Redefinitions

This test case verifies rules that need to be followed for data model redefinitions.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	
HART EDD Registration Policy	1.0	
Common Tables Specification	current	

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual and Automatic

Test Objective: Redefinition rules vary across type of redefinition intended. Below is a detailed list of redefinition rules that need to be followed across the spectrum.

Test Procedure:

- Verify BIT_ENUMERATED or ENUMERATED variables are not globally redefined at the TYPE attribute (Note: The REDEFINE must only be applied to the enumeration requiring a modification) (*TFP811_001)*
- Verify BIT_ENUMERATED and ENUMERATED variables did not add enumerations which are not defined in the current Common Tables Specification. (*TFP811_002)*
- Verify ADD is not used for a BIT_ENUMERATED or ENUMERATED variable with READ only HANDLING. (*TFP811_003)*
- Verify that Universal and Common Practice ENUMERATED and BIT_ENUMERATED variables do not DELETE its enumerations when HANDLING is READ, i.e., "Read-Only" *(Note: Enumerated / Bit Enumerated must use the DELETE function to remove any enumerations which are not supported by the device during the sending of a WRITE Command)* (*TFP811_004)*
- Verify a Variable Enumeration value is not DELETEd while HANDLING is READ *(Note: DELETE is acceptable If HANDLING is "READ & WRITE" in a standard DD and there is a Write Command applicable to the variable.)* *(TFP811_005)*
- Verify that Universal and Common Practice BIT_ENUMERATED variable does not DELETE its bit enumerations when HANDLING is READ unless the HART specification allows for truncation of bits (i.e., Command 48 Device Specific data)
- Verify ADD is not used on a variable that is not of data type BIT_ENUMERATED or ENUMERATED. (*TFP811_006)*
- Verify REDEFINE or ADD “ENUMERATOR” is not used for a variable that does not exist in the current Common Tables Specification or Standard DD

(for an IMPORT redefinition) so that *"ERROR 401: Could not find enumerator integer of VARIABLE item-name in device device-id"* is not displayed. (*TFP811_007)*

- Verify that the data used for the device id or device information (manufacturer-id, device-type device-revision, dd-revision) is less than the maximum allowable size so that *"ERROR 203: Import definition for device id is too large"* is not displayed. (*TFP811_008)*
- Verify there is no redefinition of a MIN_VALUE when MIN_VALUE does not exist in the VARIABLE's standard definition so that *"ERROR 402: Could not find MIN_VALUE integer in VARIABLE variable-name in device device-id"* is not displayed. (*TFP811_009)*
- Verify there is no redefinition of a MAX_VALUE when MAX_VALUE does not exist in the VARIABLE's standard definition so that *"ERROR 403: Could not find MAX_VALUE integer in VARIABLE variable-name in device device-id"* is not displayed. (*TFP811_010)*
- Verify the VARIABLE TYPE specifiers match between redefinition and original definition of that VARIABLE so that *"ERROR 408: Types mismatched between redefinition and original definition of VARIABLE variable-name in device device-id"* is not displayed[1]*.* (*TFP811_011)*

8.1.2 Test ID: TC_812-General Command Redefinitions

This test case verifies rules that need to be followed for Command redefinitions.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual and Automatic

Test Objective: Verify the rules for error free and best practice redefinitions of Commands, where permitted.

Test Procedure:

- Verify no mandatory attributes of Commands were deleted. (*TFP812_001)*
- Verify there are no REDEFINEd Non-Truncatable Standard Commands and the data referenced in the Standard DDs is not altered (Note: Command 3, 9 and 48 are Truncatable Commands). (*TFP812_002)*
- Verify no Universal COMMANDs are deleted. (*TFP812_003)*
- Verify all CLASS LOCAL variables use the "INFO" qualifier. (*TFP812_004)*
- Verify no Standard Command RESPONSE_CODE is DELETEd or REDEFINEd. (*TFP812_005)*
- Verify no Command attribute was REDEFINEd. "*ERROR 652: Redefinition of attribute-name in COMMAND command-name not allowed". "ERROR 653:*

---

[1] For example, this error occurs when redefining a VARIABLE type ENUMERATED and specifying an INTEGER type of VARIABLE in its place. If this change is necessary, please see the *HART EDD Registration Policy.*

Universal command command-number attribute-name attribute redefinition in COMMAND command-name not allowed." "ERROR 654: Common Practice command command-number attribute-name attribute redefinition in COMMAND command-name occurred". *(TFP812_006)*

8.1.3 Test ID: TC_813-Universal Commands Redefinitions

This test case verifies rules that need to be followed for UNIVERSAL Command redefinitions.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files.
- Open HART Test System device test logs

Test Type: Manual and Automatic

Test Objective: Verify rules are followed for error free and best practice redefinition of Universal Commands.

Test Procedure:

- Verify devices supporting 4 or more Dynamic Variables do not REDEFINE Command 3. *(TFP813_001)*
- Verify devices supporting 4 or more Dynamic Variables do not REDEFINE Command 8. *(TFP813_002)*
- Verify Command 33 (if supported) is not REDEFINEd if Command 9 is REDEFINEd. (Command 33 or 9 can only be redefined for non-FDI EDD) *(TFP813_003)*
- Verify if Command 9 REDEFINEd then number of slots must match the count determined from Maximum Number of Device Variables returned in Command 0 and the number of slots must be between 4 and 7 (for REDEFINEd, Command 9, otherwise it is 8 slots without REDEFINE). *(Note: HART 7 only: The maximum number of slots depends on the "Maximum Number of Device Variables" value returned in Command 0. No redefinition is needed if the device returns a value of 7 or higher for the Maximum Number of Device Variables in Command 0. Total count of Device Variables is "Maximum Number of Device Variables" + 1 to account for Device Variable Code 0)* *(TFP813_004)*
- Verify Command 9 is not REDEFINEd if Command 33 is REDEFINEd. (Command 33 or 9 can only be redefined for non-FDI EDD) *(Note: HART 6 device only: Maximum Number of Slots is always 4)* *(TFP813_005)*
- Verify Command TRANSACTION is not REDEFINEd when the Command TRANSACTION did not exist in the original Command definition. *(TFP813_006)*

8.1.4 Test ID: TC_814-Common Practice Commands Redefinitions

This test case verifies the rules for Common Practice Command redefinitions.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	
Common Practice Command Specification	11.0	Table 5

Prerequisites/Setup

- Open EDD source files.
- Open the HART Test System device test logs

Test Type: Manual

Test Objective: Verify rules are followed for error free and best practice redefinition of Common Practice Commands.

Test Procedure:

- Verify the Import of the Common Practice Commands uses the latest released Standard DD. *(TFP814_001)*
- Verify that if the standard Common Practice Command and its variables were REDEFINEd that the HART Protocol specifically allows for this modification (i.e., only truncatable commands may be modified.) *(TFP814_002)*
- Verify Response Codes are not redefined for any HART Protocol Standard Common Practice Commands. *(TFP814_003)*

Burst Commands:

- The “Burst” Command numbers are 103, 104, 105, 107, 108, 109. Verify the complete set (must have all commands or none) of Burst Commands are implemented. Verify against the device test logs, test is not applicable if the device does not support this feature. *(TFP814_004)*
- Verify the redefinitions of Burst Commands match the examples provided by FieldComm Group[1]. *(TFP814_005)*

Trend Commands:

- The “Trend” Command numbers are 91, 92, 93. Verify the complete set (must have all commands or none) of Trend Commands are implemented. Verify against the device test logs, test is not applicable if the device does not support this feature. *(TFP814_006)*
- Verify the redefinitions of Trend Commands match the examples provided by FieldComm Group[1]. *(TFP814_007)*

Event Notification Commands:

- The “Event Notification” Command numbers are 115, 116, 117, 118 and 119. The complete set (must have all commands or none) of Event Notification Commands were not implemented. Verify against the device test logs, test is not applicable if the device does not support this feature. *(TFP814_008)*
- Verify Burst Mode is supported when Event Notification Commands are implemented. *(TFP814_009)*

---

[1] NOTE: The Standard DD Library will have Burst, Trend, Event Notification and Condensed Status Commands included with the import of Common Practice Commands, therefore redefinitions will not be allowed.

- Verify the Variables used in Command 48 mirror the variables used in Command 115. *(TFP814_010)*

Condensed Status Commands:

- The "Condensed Status" Commands (523, 524, 525, 526, and 527) did not match the source provided by FieldComm Group[1]. Verify against the device test logs, test is not applicable if the device does not support this feature. *(TFP814_011)*
- Verify the Variables used in the Condensed Status Commands mirror the set of bytes in Command 48.[2] *(TFP814_012)*

8.1.5 Test ID: TC_815-Device Specific Command Redefinitions

This test case verifies rules for Device Specific Command redefinitions.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	
Command Summary Specification	10.0	Table 9

Prerequisites/Setup

- Open EDD source files.
- Open Field Device Specification for the device
- Open HART Test System device test logs

Test Type: Manual

Test Objective: Verify rules are followed for error free and best practice redefinition of Device Specific Commands.

Test Procedure:

- Verify Device Specific Commands supported in the DD match the *Field Device Specification* (LIT-018) belonging to the HART-Registered product. *(TFP815_001)*
- Verify Device Specific Command NUMBER is within the allowable range ("HART Command Number Partitions"). *(TFP815_002)*

---

[1] NOTE: The Standard DD Library will have Burst, Trend, Event Notification and Condensed Status Commands included with the import of Common Practice Commands, therefore redefinitions will not be allowed.

[2] Reference TS20151, Revision 11.0, Table 5. Indexes for Status Bits Found in Field Devices

9 Host Application Interface

Host applications are collections of controls that allows the device configuration and operational data to be accessed. This test case verifies the implementation of the host application interface.

9.1.1 Test ID: TC_911-Inclusion of Mandatory Menus

This test case confirms inclusion of mandatory root_menus in the DD.

Reference:

Specification	Rev.	Sections
Device Description Language Specification	14.0	13.1.1

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual

Test Objective: As per Device Description Language Specification, there is a list of root_menus which are mandatory. This test verifies inclusion of those mandatory menus.

Test Procedure:

- Verify all mandatory root_menu(s) are implemented.

  root_menu
  device_root_menu
  process_variables_root_menu
  diagnostics_root_menu (*TFP911_001)*

  The Device Description Language Specification currently requires the following root menus; however, they may be excluded from HART EDDs.

  offline_root_menu[1]
  maintenance_root_menu

---

[1] The offline_root_menu is a requirement for FDI Packages. See FCG PD10028.

10 Device to DD Comparison

Device Description Language is used to model a conformant field device. Device to DD comparison verifies items that must be consistent between a device and its DD.

10.1.1 Test ID: TC_1011-Device to DD Comparison.

This test case verifies constancy between DD and Device.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files.
- Open HART Test System device log files

Test Type: Manual

Test Objective: The use of Commands, Variables, functions etc. must correlate between device test log files and the DD. Checklist below confirms the correlation.

Test Procedure:

- Verify all Commands in this Device Revision’s DD are present in the Device having the same Device Revision. (*TFP1011_001)*
- Verify Command byte counts match the device via test log file.
  - Verify the device Command 3 byte count against the dynamic variables defined in the DD.
  - Verify the device Command 8 bytes against the dynamic variables defined in the DD.
  - Verify the device Command 9 device variables are defined in the DD.
  - Verify the device Command 9 number of slot variables match the number found in the DD (minimum 4, maximum 8).
  - Verify the device Command 48 byte count matches the response bytes in the DD. *(TFP1011_002)*
- Verify all Commands and Variables are not removed from the encoded EDD by means of pre-processor directives surrounding them (i.e. #if/#ifdef). The User Interface may use Visibility/Validity rules to conditionally show the user the pertinent information. The variables must all exist. *(TFP1011_003)*

11 Symbols Table Verification

Symbol table verification is to ensure that symbols that are supported by the field device are properly maintained. Standard symbols are controlled by FieldComm Group.

11.1.1 Test ID: TC_1111- Symbols Table Review

Test case verifies the symbols table remains consistent through the product lifecycle.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files.
- Build the symbols table[1]

Test Type: Manual

Test Objective: Verify the symbols are created correctly for the applicable Tokenizer versions.

Test Procedure:

- Verify all Commands appear in the symbols table between all applicable TOKENIZER versions. (Note: Commands must not be bypassed by pre-processor "_TOKVER_" or other directives.) *(TFP1111_001)*
- Verify all Variables appear in the symbols table between all applicable TOKENIZER versions. (Note: Variables must not be bypassed by pre-processor "_TOKVER_" or other directives.) *(TFP1111_002)*
- Verify no build errors occurred using latest released version of the Tokenizer. *(TFP1111_003)*
- Verify the item ID for an item matches between the symbol and the shape file so that "*ERROR 232: shape file item item-name is not a properly defined symbol"* is not displayed. *(TFP1111_004)*
- Verify no items have duplicate symbol IDs so that "*ERROR 508: Items item-name-1 and item-name-2 have the same symbol ID 0xsymbol-id or symbol-id"* is not displayed. *(TFP1111_005)*
- Verify the Symbol(s) were created so that "*ERROR 517: Item item-name was imported but is not in a symbol file"* is not displayed. *(TFP1111_006)*
- Verify the Item ID is not redefined for an existing item so that "*ERROR 519: file-name, line line-number: Item item-name already exists with Item ID 0xsymbol-name or symbol-name, cannot be redefined*" is not displayed. *(TFP1111_007)*

---

[1] Note: User option *-a* must be used to ensure that all symbols are declared in the symbol file. FieldComm Group creates the final Symbol table for all EDDs.

12 Registration Type Specific Rules

These test points are applied based on the type of registration. The test case verifies the rules for a New Field Device DD, Device Revision DD, or EDD Revision.

12.1.1 Test ID: TC_1211-Registration Type Specific Rules

Test case verifies the updates are consistent with field device revision rules of the *Command Summary Specification*.

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	
Command Summary Specification	10.0	Section 6

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual

Test Objective: Compare the source code to the previously registered files.

Test Procedure:

Device Revision DD or DD Revision:

- Verify Symbol ID remain identical between revisions. (*TFP1211_001)*
- Verify Commands or Data references (LIST, Value Array, Reference Array or other Variable reference which was previously referenced in a command) are not deleted when comparing to the previous source code revision. (*TFP1211_002)*

DD Revision:

- Verify no new Variables, Enumerations, or Commands were added when compared to the last registered EDD and the corresponding registered device. (i.e. All DD Revisions for a single Device Revision must work with all devices in that Device Revision. A DD Revision cannot be registered if it does not work with all devices of a specific Device Revision.) (*TFP1211_003)*

New Field Device DD or Device Revision:

- Verify the device is *HART-Registered* or currently under test. (*TFP1211_004)*

13 Private Label or Rebranded Products

13.1 Test ID: TC_1311-Private Labelling Requisites

Reference:

Specification	Rev.	Sections
HART EDD Registration Policy	1.0	

Prerequisites/Setup

- Open EDD source files.

Test Type: Manual

Test Objective: Validate a Private Label or Rebranded product DD is conformant to the latest HART Protocol and EDDL requirements, as well as a true copy of the original product.

Test Procedure:

- Verify the Private Label device is *HART-Registered* or currently under test. (*TFP1311_001)*
- Verify the Private Label DDs contains no deviations from the original product's DD, other than identifying information (device_type, private_label_distributor, images)[1]. (*TFP1311_002)*

---

[1] If a product is tested as a Private Label or Rebranded device and differences are discovered, this disqualifies the product from the Private Label or Rebrand registration process path and will be treated as a new unique product instead.

Annex A Tokenizer Errors

Below is a list of Tokenizer errors that will cause a test failure. These errors must be fixed by the developer prior to submission to FieldComm Group.

Tokenizer ERROR List:

- ERROR 100: Memory exhausted.
- ERROR 101: File system error; seek failed for file file-name.
- ERROR 102: File system error; write failed for file file-name.
- ERROR 103: File system error; read failed for file file-name.
- ERROR 104: File system error; stat failed for file file-name.
- ERROR 105: Cannot open file file-name
- ERROR 106: Configuration file file-name contains input files.
- ERROR 107: Too many recursions in configuration files.
- ERROR 108: Multiple default configuration files specified.
- ERROR 109: No parameter for command-line option option-name.
- ERROR 110: Unknown command-line option option-name.
- ERROR 111: Cannot change default configuration file within configuration file file-name.
- ERROR 114: Unable to open Tokenizer setup file: file-name.
- ERROR 115: Input file: file-name is same as Output file: file-name.
- ERROR 202: Attributes for item item-name too large.
- ERROR 205: Device Directory is too large.
- ERROR 206: Block Directory for block block-name is too large.
- ERROR 210: DDOD binary file file-name has incompatible revision number revision-number.
- ERROR 211: DDOD binary file file-name has incompatible format.
- ERROR 212: File file-name is not a valid DDOD binary.
- ERROR 220: Enumerations specifier has no enumerators.
- ERROR 221: Elements specifier has no elements.
- ERROR 222: Members specifier has no members.
- ERROE 223: Response codes specifier has no response codes.
- ERROR 225: Invalid dictionary string [section-number, string-id] dictionary-string.
- ERROR 227: Variable variable-name: multibyte variable cannot be used as an index in command command-number.
- ERROR 228: Variable variable-name: cannot be used as an index in command command-number.
- ERROR 229 Variable variable-name, used in command command-number, is neither parameter nor local variable.
- ERROR 231: variable-name, line line-number: Shape file syntax error.

- ERROR 232: file-name, line line-number: shape file item item-name is not a properly defined symbol.
- ERROR 305: file-name, line line-number: Invalid source character: character.
- ERROR 306: file-name, line line-number: Invalid preprocessor directive.
- ERROR 307: file-name, line line-number: String contains duplicate country codes.
- ERROR 308: file-name, line line-number: syntax-error appears more than once.
- ERROR 309: file-name, line line-number: syntax error OR file-name, line line-number: stack overflow.
- ERROR 313: Invalid reference
- ERROR 311: file-name, line line-number: Invalid escape sequence.
- ERROR 317: file-name, line line-number: Invalid dictionary string name string.
- ERROR 321: file-name, line line-number: Invalid imported device description.
- ERROR 323: file-name, line line-number: method has incompatible built-in use.
- ERROR 325: file-name, line line-number: invalid index-number index.
- ERROR 327: file-name, line line-number: String language/country code is missing closing delimiter.
- ERROR 328: file-name, line line-number: String contains bad language/country code.
- ERROR 329: file-name, line line-number: String contains invalid old country code.
- ERROR 405: Redefinition error in file-name, line line-number: Could not find element integer of item item-name in device device-id.
- ERROR 411: Redefinition error in file-name, line line-number Item
- ERROR 414: Could not find TRANSACTION transaction-number of COMMAND command-name in device device-id.
- ERROR 502: file-name, line line-number: Standard item item-name with 0xsymbol-name, or symbol-name is not in the standard symbol table.
- ERROR 503: Item item-name defined in file-name, line-number was previously defined in file file-name, line-number.
- ERRPOR 505: Cannot open symbol table file file-name
- ERROR 506: file-name, line line-number: Syntax error.
- ERROR 507: file-name, line line-number: Invalid flag flag-name for item item-name.
- ERROR 510: Item item-name is defined but not used.
- ERROR 534: file-name, line line-number: Refresh relation <name>, variable list as dominant in a relation is deprecated.
- ERROR 602: item-name-1 appears more than once in item-name-2.
- ERROR 603: item-name-1 is the wrong type of item to be used in ddl construct item-name-2 as a attribute-name value. *)*
- ERROR 608: item-name imported multiple times.
- ERROR 611: Undefined attribute from item-name was deleted.
- ERROR 613: Invalid flags defined for menu item menu-item in MENU menuname.
- ERROR 614: ddl-construct item-name is always invalid.
- ERROR 615: The CLASS for item-name is incorrect.
- ERROR 617: The size for VARIABLE variable-name is too large. or 617 The size for VARIABLE variable-name is too small.
- ERROR 620: MEMBERS of ddl-construct item-name have different types.
- ERROR 621: ELEMENTS in ddl-construct item-name have different types.
- ERROR 622: Invalid format-string-element in DISPLAY_FORMAT string for variable name or 622 Invalid format-string-element in EDIT_FORMAT string for variable variable-name.
- ERROR 624: DEFAULT not last CASE in SELECT.

- ERROR 629: Invalid reference found involving item-name at line line-number of file-name ID 0xsymbol-name, type item-type.
- ERROR 630: item-name has No members/elements Defined.
- ERROR 631: METHOD method-name is used by VARIABLE variable-name as a pre/post read/write action but contains non-scaling built-in routines.
- ERROR 637: Multiple transactions are defined for COMMAND command-name.
- ERROR 640: COMMAND command-number is RESERVED and its usage in command name is not permitted.
- ERROR 642: COMMAND command-name TRANSACTION transaction-number REQUEST data-item is invalid. Or COMMAND command-name TRANSACTION transaction-number REPLY data-item is invalid.
- ERROR 644: construct-identifier value response-code-value is greater than maximum maximum-value.
- ERROR 647: COMMAND command-name TRANSACTION transaction-number REQUEST data item masks leave bit stream gap or COMMAND command-name TRANSACTION transaction-number REPLY data item masks leave bit stream gap.
- ERROR 649: COMMAND command-name TRANSACTION transaction-number REQUEST has no WRITE operation in the data items. Or COMMAND command-name TRANSACTION transaction-number REPLY has no READ operation in the data items.
- ERROR 651: COMMAND command-name TRANSACTION transaction-number REQUEST variable variable-name is wrong type for a data item mask or 651 COMMAND command-name TRANSACTION transaction-number REPLY variable variable-name is wrong type for a data item mask.
- ERROR 657: COMMAND command-name TRANSACTION transaction-number data items improperly qualified.
- ERROR 659: Functional class for value hex-value or decimal-value in VARIABLE variable-name is incorrect.
- ERROR 660: Bit-enumerated value hex-value or decimal-value specified in VARIABLE variable-name is incorrect.
- ERROR 662: Index VARIABLE variable-name references undefined ARRAY array-name.

- ERROR 664: Item item-name is used as construct-type in construct-type item-name but is undefined.
- ERROR 666: construct-type item-name: attribute-type is invalid attribute for HART device.

- ERROR 667: Variable variable-name: status is incorrect bit-enumerator status class for HART device.
- ERROR 669: VARIABLE variable-name has CONSTANT_UNITS deleted on import, which may affect interoperability.
- ERROR 674: Attempt to delete a mandatory item: <name>.
- ERROR 701: file-name, line line-number: HART method method-name encountered and cannot be converted.
- ERROR 678: file-name, line line-number: Item <name> (<num>) is used in a reference and is not defined.
- ERROR 679: COMMAND <name> TRANSACTION <num> request item #<n> qualifiers do not match the reply qualifiers.
- ERROR 680: file-name, line line-number: Item used as INDEX is not type INDEX.
  ERROR 703: file-name, line line-number: Path conditional could not be
  resolved for <name>.

- ERROR 704: Variable <name>: TIME_SCALE value is not permitted.
- ERROR 707: 707 Method: method-name, line line-number: <error-message>.
- ERROR 709: 709 Attempt to output duplicate Symbol Numbers <num>.
- ERROR 710: 710 String length (<length>) may not fit in some hosts.