



## Falcon UniPro Stimulus Builder User Manual

Copyright © Protocol Insight. All rights reserved. Licensed software products are owned by Protocol Insight or its suppliers, and are protected by national copyright laws and international treaty provisions.

Protocol Insight products are covered by U.S. and foreign patents, issued and pending. Information in this manual supersedes all previously published material. Details, specifications and pricing subject to change.

Protocol insight is a registered trademark of Protocol Insight, LLC.

MIPI and the MIPI logo are a licensed trademark of the MIPI Alliance.

UFS and UFS Logo are a trademark of the Universal Flash Storage Association

JEDEC® and the JEDEC logo are registered trademarks of JEDEC Solid State Technology Association.

Contact Protocol Insight at:

[sales@protocolinsight.com](mailto:sales@protocolinsight.com)

[support@protocolinsight.com](mailto:support@protocolinsight.com)

[www.protocolinsight.com](http://www.protocolinsight.com)



The Protocol Insight® UniPro Stimulus Builder™ tool is available with the Falcon G350 and G450 software. It allows the creation of custom test cases to test corner cases or introduce errors for margin, stress, or error recovery testing. UniPro Stimulus Builder allows complex stimulus test creation, with virtually unlimited transactions, device control, and command packet construction. Global variables and run conditions are used to create test case conditions.

Once a custom test case is created it can be executed from the CTS Configuration or Exerciser + Analyzer Configuration window separately or together with the pre-defined test cases, and can be run using the Test Run Control and Test Run Order stress testing features.

## Contents

UniPro Stimulus Builder Overview .....	1
Test Case ID .....	2
Launching Stimulus Builder .....	3
UniPro Stimulus Builder menu .....	3
File .....	3
Help .....	4
Scripts .....	5
Variables .....	26
Global Variable .....	26
Payload .....	27
Appendix A: Syntax Checker Error Messages .....	28
Appendix B: Known Limitations .....	29



## UniPro Stimulus Builder Overview

Figure 3 shows the UniPro Stimulus Builder template. By right clicking on headers the user can add Test Case ID, Run Conditions which contain Global Variables, and Transactions which can contain Scripts and L4 Payloads.

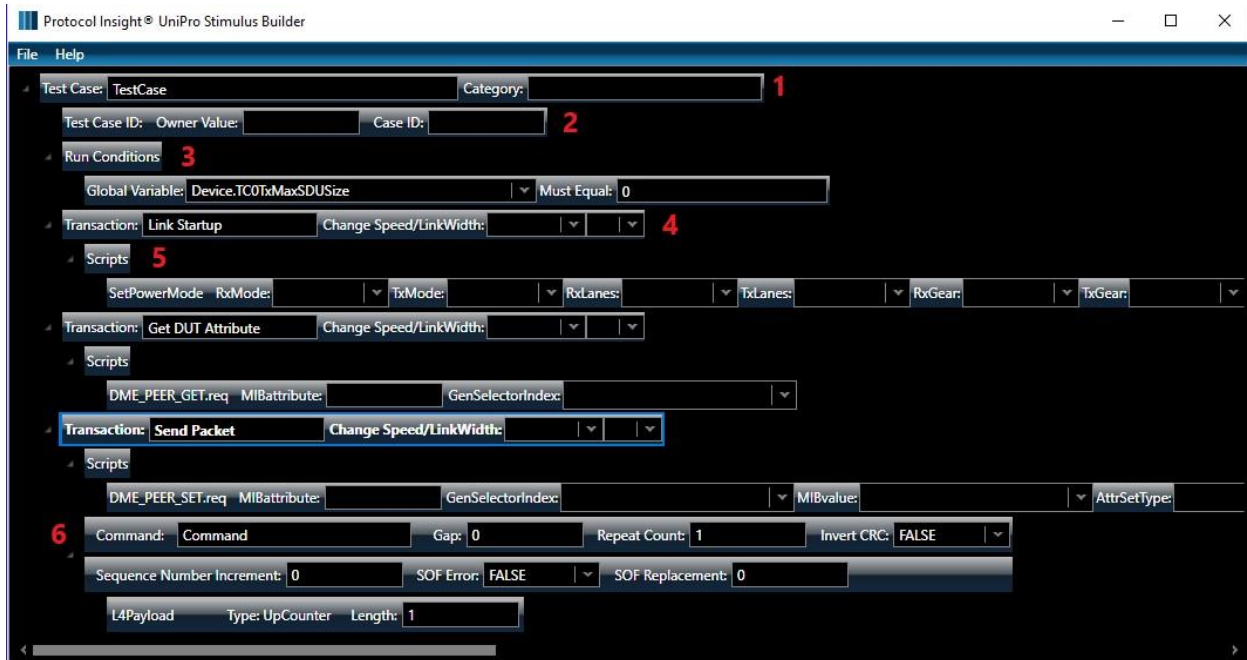


Figure 3: Main Sections of a Test Case

- Test case:** The name of the test case, which is also the file name the Test Case gets saved as.
  - Category:** this is the Folder under <User>\Documents\Protocol Insight\Falcon\CustomTests that the test case is saved into. This is also the Category in the Tests list that the test case will be listed under.
- Test Case ID:** The test case ID can be used to inject a DME\_PEER\_SET.req with the MIBAttribute set to the Owner value and the MIBvalue set to Case ID.
- Run Conditions:** This is used to decide if the Test Case will be executed or not. All of the Run Conditions must be true for the test case to run. To Create a Run Condition select a Global Variable from the pull down then set the Value the Global Variable MUST be for the test case to run. See the [GLOBAL VARIABLE](#) section for more details.
- Transaction:** A Transaction is typically a payload or a script.
  - Name:** Create a name that helps you understand what is in the transaction.
  - Change Speed/LinkWidth:** If this is left blank the Transaction is executed at the speed and Linkwidth based on the current iteration of the Test Matrix. If these fields are set then ONLY this transaction is executed at the speed and/or Linkwidth that is set.
- Scripts:** A Command is sent to the DUT, the user defines every byte that is sent. See the [Scripts](#) section for more details.



6. **Payload:** A payload is a L4 data packet that is sent from the tester to the DUT. A number of parameters can be set in the packet header to affect how the packet is sent. See the [Payload](#) section for more details.

## Test Case ID

To indicate the beginning of each test case on the bus, testing is initiated with a PACP\_SET\_req marker packet, followed by the packet(s) necessary to implement the test. The marker packet contains a value that uniquely identifies the test case:

- The PACP\_SET\_req **MIBattribute** value (Owner Value field in Stimulus Builder) identifies the author of the test case. The predefined ranges used are listed below.
  - **0xFFFF** – Test case authored by Protocol Insight
  - **0xFF00-0xFFFE** – This range reserved
  - **0xF000-0xFEFF** – Range available for customer defined test cases.
- The **MIBvalue** (Case ID field in Stimulus Builder) identifies the specific test being executed:
  - **Domain** (first 8 bits) – Used to define a category of test case for the given owner. Existing Protocol Insight domains are:
    - **01** - CTS specific test cases
    - **02** - Power mode tests
  - **Case ID** (last 24 bits) – Identifies a specific test case within a given domain. Protocol Insight test cases follow a major, minor, revision style using 8 bits for each.

For example, the CTS test case 1.1.4 Invalid Configuration Request written by Protocol Insight would have a complete ID of 0x01010104.



## Launching Stimulus Builder

To run the UniPro Stimulus Builder, from the main menu of the Falcon software select Tools→UniPro Stimulus Builder. This will launch Stimulus Builder in a separate window.

## UniPro Stimulus Builder menu

### File

#### File→New

Creates a blank test case template.

#### File→Open

Open existing custom test cases located in <User>\Documents\Protocol Insight\Falcon\CustomTests.

To open Protocol Insight pre-defined tests to modify and save as custom test cases, navigate to C:\Program Files\Protocol Insight\Falcon\Stimulus\Data\TestCases.

#### File→Save and Save As

Saves custom test cases in <User>\Documents\Protocol Insight\Falcon\CustomTests as .pis (Protocol Insight Stimulus) files.

Note: test cases are saved in a subdirectory under CustomTests which appear in Stimulus Builder as a Category. Categories are displayed in the Tests section of the Configuration window, as well as the File -> Save As dialog.

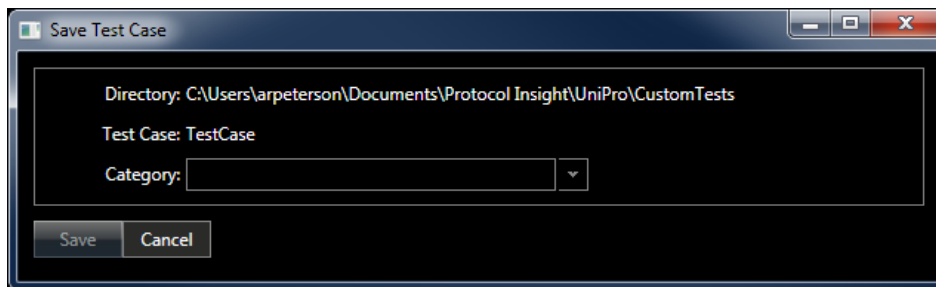


Figure 1: Save Test Cases

Categories are sub-directories under <User>\Documents\Protocol Insight\Falcon\CustomTests and are displayed in the Tests section of the Configuration window. New categories can be created by typing the new category name into the Category field in the Save dialog before saving, new categories can also be created by manually adding new sub-directories in <User>\Documents\Protocol Insight\Falcon\CustomTests with File Explorer.

See the Falcon Series User Manual for more information.



Once subdirectories have been created, they will be visible from the drop-down arrow.

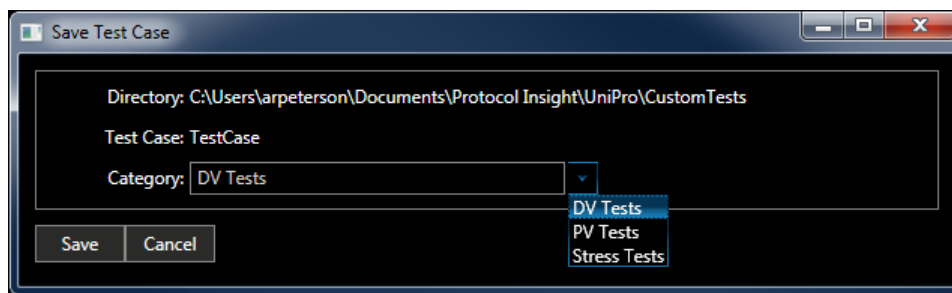


Figure 2: Save Test Cases Category

File→Run Syntax Checker

Runs the test case through a basic Syntax check. Useful for creating custom test cases.

File→Exit

Exits the Stimulus Builder tool. Will prompt to save the current test case if any changes are detected.

[Help](#)

Help→UniPro Stimulus Builder User Manual

Opens a PDF version of this document.



## Scripts

Scripts can be added to perform different tasks; to add a script, right click the Transaction header and select Add Scripts. Right click the Scripts header and select Add Script and choose a script as shown in Figure 4.

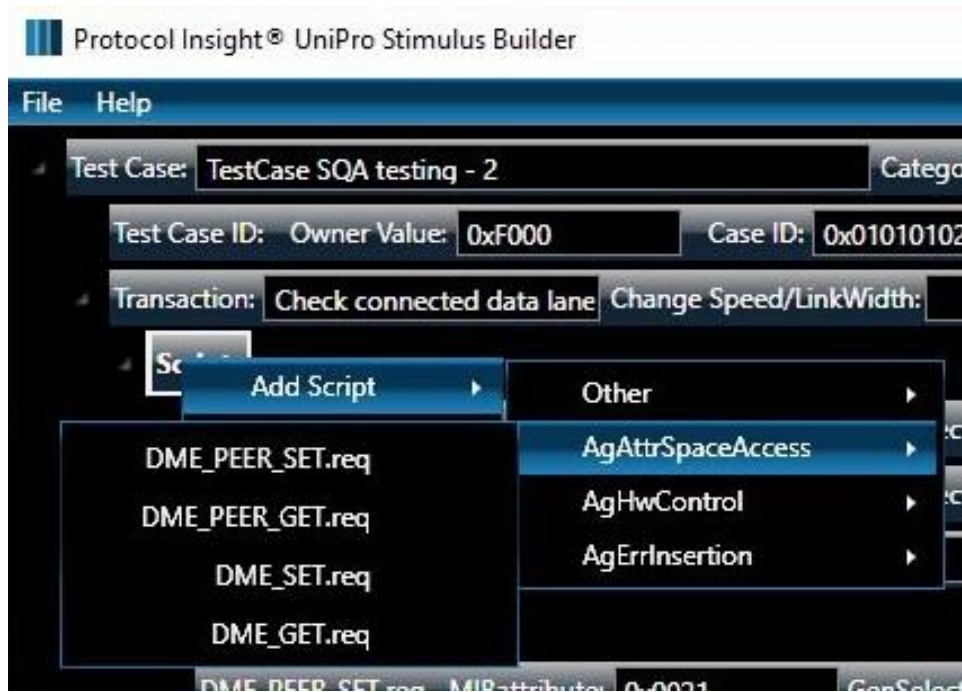


Figure 4: Right Click Menu of Script Menu

The following scripts are currently supported:

### reBootUniPro



Figure 5: ReBootUniPro

This function reboots the UniPro stack by resetting both devices to re-initiate link startup.



## reBootUniProOnce

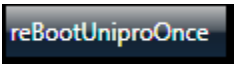


Figure 6: ReBootUniProOnce

This function reboots the UniPro stack by resetting the A-side device to re-initiate link startup between the devices.

## ResetPacketChanger



Figure 7: ResetPacketChanger

This function causes the internal Packet Changer tool to be reset to its default state, clearing any changes or errors that have been inserted.

## SetPowerMode

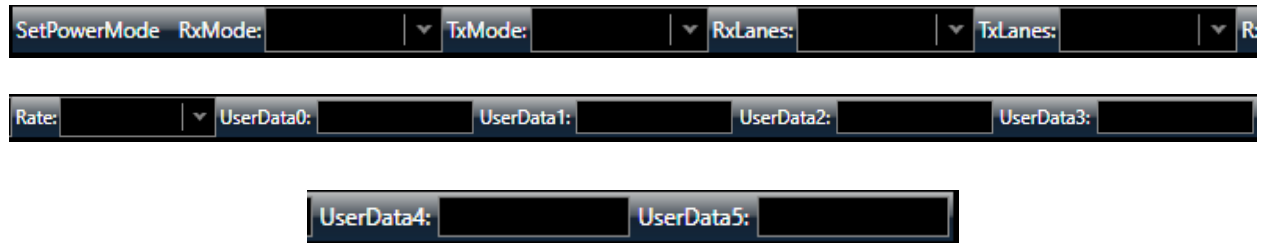


Figure 8: SetPowerMode

This function is used to change the power mode of the link.

### Parameters

RxMode:

Input type: Integer

Range: 1- Fast, 2 – Slow, 4 – FastAuto, 5 – SlowAuto, 7 - Unchanged

Description: Specifies the requested power mode





## TxMode:

Input type: Integer

Range: 1- Fast, 2 – Slow, 4 – FastAuto, 5 – SlowAuto, 7 - Unchanged

Description: Specifies the requested power mode

## RxLanes:

Input type: Integer

Range: 1-4

Description: Specifies the number of Rx lanes

## TxLanes:

Input type: Integer

Range: 1-4

Description: Specifies the number of Tx lanes

## RxGear:

Input type: Integer

Range: 1-7

Description: Specifies the Gear to set for Rx

## TxGear:

Input type: Integer

Range: 1-7

Description: Specifies the Gear to set for Tx

## RxTermination:

Input type: Integer

Range: ENABLED or DISABLED

Description: Specifies whether the Rx Terminations are enabled



## TxTermination:

Input type: Integer

Range: ENABLED or DISABLED

Description: Specifies whether the Tx Terminations are enabled

## Scrambling:

Input type: Integer

Range: ENABLED or DISABLED

Description: Specifies whether scrambling is enabled for the link

## Rate:

Input type: Integer

Range: A or B

Description: Specifies what Rate series the link uses

## UserData0 – UserDat5:

Input type: Integer (specified in Specification)

Range: [VARIABLE]

Description: PA\_PWRModeUserData parameters 0-5



## WaitForTime



Figure 9: WaitForTime

This function inserts a wait state into the link for the specified time in ms.

### Parameters

#### After:

Input type: Integer

Range: 0 – 65,535

Description: Specifies the number of ms to wait

## WaitL2Empty



Figure 10: WaitL2Empty

This function causes the Exerciser to wait a specified time (if provided), then to wait for the L2 layer of the UniPro stack to be empty (or until the specified TimeOut expires) before proceeding.

### Parameters

#### TimeOut:

Input type: Integer

Range: 0 – 65,535

Description: Specifies a fixed Timeout in mS to wait for L2 layer depletion before proceeding. Defaults to 500 mS.

#### WaitTimeAfter:

Input type: Integer

Range: 0 – 65,535

Description: Specifies the number of mS to wait before sampling the L2 layer for depletion. Defaults to 0.



## DME\_PEER\_SET.req



Figure 11: DME\_PEER\_SET.req

This function is used to set the MIBvalue value of a specific Attribute of the peer Device, identified by MIBattribute and, if relevant, the GenSelectorIndex. The GenSelectorIndex is interpreted either as a data PHY Lane index or CPort index depending on the Attribute. For Attributes not associated with a GenSelectorIndex#, the GenSelectorIndex is ignored.

### Parameters:

#### MIBattribute:

Input type: hex value as specified in the Specification

Range: [0x0000 – 0x7FFF]

Description: Indicates the MIB Attribute.

#### GenSelectorIndex:

Input type: Drop-down selection of device scripts

Range: Choice of 32 scripts

- Device.TC0TxMaxSDUSize
- Device.TestFeatureCPortRange
- Device.BNumberLU
- Device.BNumberWLU
- Device.BNumberLUPlusbNumberWLU
- Device.IManufacturerName
- Device.IProductName
- Device.IOemID
- Device.ISerialNumber
- Device.IProductRevisionLevel
- Device.UniqueValue
- Device.BInitActiveICCLLevel
- Device.BDeviceRTTCap
- Device.BMaxInBufferSize
- Device.BMaxOutBufferSize
- Device.BConfigDescrLockEquals0
- Device.BBootEnableEquals1
- Device.BInitPowerModeEquals1
- Device.BOutOfOrderDataEnHasBeenWritten
- Device.BMaxDataOutSize
- Device.BRefClkFreqHasBeenWritten



Device.BConfigDescrLockHasBeenWritten  
Device.EnableWriteOnceAttributeTestCases  
Device.ImmedSupported  
Device.TC0TxMaxSDUSizeX13  
Device.TC0TxMaxSDUSizeX37  
Device.TC0TxMaxSDUSizeX128  
Device.TC0TxMaxSDUSizeX240  
Device.FirstTestFeatureCPort  
Device.FourthTestFeatureCPort  
Device.SecondTestFeatureCPort  
Device.ThirdTestFeatureCPort

Description: Indicates the targeted M-PHY data lane or CPort when relevant

MIBValue:

Input type: Drop-down selection of device scripts

Range: Choice of 32 scripts

Device.TC0TxMaxSDUSize  
Device.TestFeatureCPortRange  
Device.BNumberLU  
Device.BNumberWLU  
Device.BNumberLUPlusbNumberWLU  
Device.IManufacturerName  
Device.IProductName  
Device.IOemID  
Device.ISerialNumber  
Device.IProductRevisionLevel  
Device.UniqueValue  
Device.BInitActiveICCLLevel  
Device.BDeviceRTTCap  
Device.BMaxInBufferSize  
Device.BMaxOutBufferSize  
Device.BConfigDescrLockEquals0  
Device.BBootEnableEquals1  
Device.BInitPowerModeEquals1  
Device.BOutOfOrderDataEnHasBeenWritten  
Device.BMaxDataOutSize  
Device.BRefClkFreqHasBeenWritten  
Device.BConfigDescrLockHasBeenWritten  
Device.EnableWriteOnceAttributeTestCases  
Device.ImmedSupported  
Device.TC0TxMaxSDUSizeX13  
Device.TC0TxMaxSDUSizeX37  
Device.TC0TxMaxSDUSizeX128  
Device.TC0TxMaxSDUSizeX240



Device.FirstTestFeatureCPort  
 Device.FourthTestFeatureCPort  
 Device.SecondTestFeatureCPort  
 Device.ThirdTestFeatureCPort

Description: Indicates value of the MIB Attribute. The MIB value cannot exceed 32 bits.

**AttrSetType:**

Input type: Integer

Range: [0 - 1] 0 indicates NORMAL, 1 indicates STATIC

Description: Selects whether Attribute value is set directly (normal) or static value is set.

Note: dropdown list has values of NORMAL and STATIC available.

**CnfRequired:**

Input type: Integer

Range: [0-1] 0 indicates not required, 1 indicates required.

Description: Confirmation required for the request or not.

**DME\_PEER\_GET.req**



Figure 12: DME\_PEER\_GET.req

This function is used to request information from a specific Attribute, in the peer Device, identified by MIBAttribute and, if relevant, the GenSelectorIndex. The GenSelectorIndex shall be interpreted either as a data PHY Lane index or CPort index depending of the Attribute. For Attributes not associated with a GenSelectorIndex, the GenSelectorIndex shall be ignored.

**Parameters**

**MIBAttribute**

Input type: Hex value or Attribute name as specified in Specification

Range [0x0000 - 0x7FFF]



Description: The address of the MIB Attribute

GenSelectorIndex

Input type: Drop-down selection of device scripts

Range: Choice of 32 scripts

- Device.TC0TxMaxSDUSize
- Device.TestFeatureCPortRange
- Device.BNumberLU
- Device.BNumberWLU
- Device.BNumberLUPlusbNumberWLU
- Device.IManufacturerName
- Device.IProductName
- Device.IOemID
- Device.ISerialNumber
- Device.IProductRevisionLevel
- Device.UniqueValue
- Device.BInitActiveICCLLevel
- Device.BDeviceRTTCap
- Device.BMaxInBufferSize
- Device.BMaxOutBufferSize
- Device.BConfigDescrLockEquals0
- Device.BBootEnableEquals1
- Device.BInitPowerModeEquals1
- Device.BOutOfOrderDataEnHasBeenWritten
- Device.BMaxDataOutSize
- Device.BRefClkFreqHasBeenWritten
- Device.BConfigDescrLockHasBeenWritten
- Device.EnableWriteOnceAttributeTestCases
- Device.ImmedSupported
- Device.TC0TxMaxSDUSizeX13
- Device.TC0TxMaxSDUSizeX37
- Device.TC0TxMaxSDUSizeX128
- Device.TC0TxMaxSDUSizeX240
- Device.FirstTestFeatureCPort
- Device.FourthTestFeatureCPort
- Device.SecondTestFeatureCPort
- Device.ThirdTestFeatureCPort

Description: Indicates the targeted M-PHY data lane or CPort when relevant

**DME\_SET.req**



Figure 13: DME\_SET.req

This function is used to set the value of a specific Attribute identified by MIBAttribute and, if relevant, the GenSelectorIndex. The GenSelectorIndex is interpreted either as a data PHY Lane index or Cport index depending of the Attribute. For Attributes not associated with a GenSelectorIndex, the GenSelectorIndex is ignored.

Parameters:

MIBAttribute:

Input type: Hex value or Attribute name as specified in Specification

Range: [0x0000 - 0x7FFF]

Description: The address of the MIB Attribute

GenSelectorIndex:

Input type: Drop-down selection of device scripts

Range: Choice of 32 scripts

- Device.TCOTxMaxSDUSize
- Device.TestFeatureCPortRange
- Device.BNumberLU
- Device.BNumberWLU
- Device.BNumberLUPlusbNumberWLU
- Device.IManufacturerName
- Device.IProductName
- Device.IOemID
- Device.ISerialNumber
- Device.IProductRevisionLevel
- Device.UniqueValue
- Device.BInitActiveICCLLevel
- Device.BDeviceRTTCap
- Device.BMaxInBufferSize
- Device.BMaxOutBufferSize
- Device.BConfigDescrLockEquals0
- Device.BBootEnableEquals1
- Device.BInitPowerModeEquals1
- Device.BOutOfOrderDataEnHasBeenWritten
- Device.BMaxDataOutSize
- Device.BRefClkFreqHasBeenWritten





Device.BConfigDescrLockHasBeenWritten  
Device.EnableWriteOnceAttributeTestCases  
Device.ImmedSupported  
Device.TC0TxMaxSDUSizeX13  
Device.TC0TxMaxSDUSizeX37  
Device.TC0TxMaxSDUSizeX128  
Device.TC0TxMaxSDUSizeX240  
Device.FirstTestFeatureCPort  
Device.FourthTestFeatureCPort  
Device.SecondTestFeatureCPort  
Device.ThirdTestFeatureCPort

Description: Indicates the targeted M-PHY data lane or CPort when relevant

MIBvalue:

Input type: Drop-down selection of device scripts

Range: Choice of 32 scripts

Device.TC0TxMaxSDUSize  
Device.TestFeatureCPortRange  
Device.BNumberLU  
Device.BNumberWLU  
Device.BNumberLUPlusbNumberWLU  
Device.IManufacturerName  
Device.IProductName  
Device.IOemID  
Device.ISerialNumber  
Device.IProductRevisionLevel  
Device.UniqueValue  
Device.BInitActiveICCLLevel  
Device.BDeviceRTTCap  
Device.BMaxInBufferSize  
Device.BMaxOutBufferSize  
Device.BConfigDescrLockEquals0  
Device.BBootEnableEquals1  
Device.BInitPowerModeEquals1  
Device.BOutOfOrderDataEnHasBeenWritten  
Device.BMaxDataOutSize  
Device.BRefClkFreqHasBeenWritten  
Device.BConfigDescrLockHasBeenWritten  
Device.EnableWriteOnceAttributeTestCases  
Device.ImmedSupported  
Device.TC0TxMaxSDUSizeX13  
Device.TC0TxMaxSDUSizeX37  
Device.TC0TxMaxSDUSizeX128  
Device.TC0TxMaxSDUSizeX240



Device.FirstTestFeatureCPort  
 Device.FourthTestFeatureCPort  
 Device.SecondTestFeatureCPort  
 Device.ThirdTestFeatureCPort

Description: The value of the MIB Attribute. The MIB value size cannot exceed thirty-two bits

## DME\_GET.req



Figure 14: DME\_GET.req

This function is used to get the value of a specific Attribute identified by MIBAttribute and, if relevant, the GenSelectorIndex. The GenSelectorIndex is interpreted either as a data PHY Lane index or Cport index depending of the Attribute. For Attributes not associated with a GenSelectorIndex, the GenSelectorIndex is ignored.

### Parameters

#### MIBAttribute :

Input type: Hex value or Attribute name as specified in Specification

Range [0x0000 - 0x7FFF]

Description: The address of the MIB Attribute

#### GenSelectorIndex:

Input type: Drop-down selection of device scripts

Range: Choice of 32 scripts

- Device.TCOTxMaxSDUSize
- Device.TestFeatureCPortRange
- Device.BNumberLU
- Device.BNumberWLU
- Device.BNumberLUPlusbNumberWLU
- Device.IManufacturerName
- Device.IProductName
- Device.IOemID
- Device.ISerialNumber
- Device.IProductRevisionLevel
- Device.UniqueValue
- Device.BInitActiveICCLLevel



Device.BDeviceRTTCap  
Device.BMaxInBufferSize  
Device.BMaxOutBufferSize  
Device.BConfigDescrLockEquals0  
Device.BBootEnableEquals1  
Device.BInitPowerModeEquals1  
Device.BOutOfOrderDataEnHasBeenWritten  
Device.BMaxDataOutSize  
Device.BRefClkFreqHasBeenWritten  
Device.BConfigDescrLockHasBeenWritten  
Device.EnableWriteOnceAttributeTestCases  
Device.ImmedSupported  
Device.TC0TxMaxSDUSizeX13  
Device.TC0TxMaxSDUSizeX37  
Device.TC0TxMaxSDUSizeX128  
Device.TC0TxMaxSDUSizeX240  
Device.FirstTestFeatureCPort  
Device.FourthTestFeatureCPort  
Device.SecondTestFeatureCPort  
Device.ThirdTestFeatureCPort

Description: Indicates the targeted M-PHY data lane or CPort when relevant

## DME\_HIBERNATE\_ENTER.req

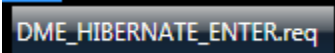


Figure 15: DME\_HIBERNATE\_ENTER.req

This function is used to put all the layers in Hibernate state.

## DME\_HIBERNATE\_EXIT.req

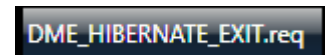


Figure 16: DME\_HIBERNATE\_EXIT.req

This function is used to wake all the layers from the Hibernate state.

## ERR\_IGNORE\_PWR\_REQ

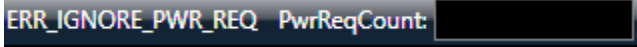


Figure 17: ERR\_IGNORE\_PWR\_REQ

This function causes PACP\_Pwr\_Req frames to be ignored.

Parameters:

PwrReqCount: - number of PACP\_Pwr\_Req frames that are to be ignored

Input type: Hex value

## ERR\_IN\_AFC



Figure 18: ERR\_IN\_AFC

This function defines the errors that are to be inserted into an AFC Frame.

Parameters:

CRC: -inverts CRC of AFC

CREQ\_BIT: -sets CReq Bit of AFC

RSVD\_BITS: -inverts reserved bits in AFC

INCR\_SEQ\_NUM: -increases the sequence number in AFC by 1

DECR\_SEQ\_NUM: -decreases the sequence number in AFC by 1

TC0: -replaces TC0 by TC1

SYMB: -results in symbol error in AFC

DISP: -results in disparity error in AFC

CREDIT: -followed by 8 bit value in hex with which AFC is to be replaced

REPLACE: -followed by 3 bit value in hex with which AFC is to be replaced

EXTRASYMBOL: -results in extra symbol in AFC



## ERR\_IN\_PACP\_CAP\_IND



Figure

19: ERR\_IN\_PACP\_CAP\_IND

This function defines the errors that are to be inserted into a PACP\_CAP\_ind Frame. Currently, only SKIP is supported.

### Parameters:

- CRC: -inverts CRC of PACP Frame
- RSVD\_BITS: -inverts Reserved bits of PACP Frame
- FUNC\_ID: -increases the function id by 1 of PACP Frame
- SYMB: -results in symbol error in PACP Frame
- DISP: -results in disparity error in PACP Frame
- SKIP: -results in not sending PACP\_CAP\_ind Frame



## ERR\_IN\_PACP\_REQ



Figure

20: ERR\_IN\_PACP\_REQ

This function defines the errors that are to be inserted into a PACP\_REQ Frame.

Parameters:

- CRC: -inverts CRC of PACP Frame
- RSVD\_BITS: -inverts Reserved bits of PACP Frame
- FUNC\_ID: -increases the function id by 1 of PACP Frame
- SYMB: -results in symbol error in PACP Frame
- DISP: -results in disparity error in PACP Frame

## ERR\_INJECT\_AFC



Figure 21: ERR\_INJECT\_AFC

This function defines the injection of an AFC Frame along with errors that are to be inserted in it.

Parameters:

- CRC: -inverts CRC of AFC
- CREQ\_BIT: -sets CReq Bit of AFC
- RSVD\_BITS: -inverts reserved bits in AFC
- INCR\_SEQ\_NUM: -increases the sequence number in AFC by 1



- DECR\_SEQ\_NUM: -decreases the sequence number in AFC by 1
- TC0: -replaces TC0 by TC1
- SYMB: -results in symbol error in AFC
- DISP: -results in disparity error in AFC
- CREDIT: -followed by 8 bit value in hex with which AFC is to be replaced
- REPLACE: -followed by 3 bit value in hex with which AFC is to be replaced
- EXTRASYMBOL: -results in extra symbol in AFC

## ERR\_REPLACE\_AFC\_WITH\_NAC

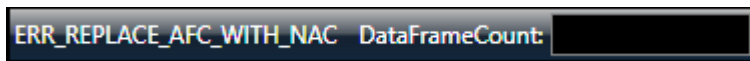


Figure 22: ERR\_REPLACE\_AFC\_WITH\_NAC

This function defines the data frame count after which NAC Frames should replace AFC Frames.

### Parameters:

#### DataFrameCount:

Input type: Hex value

Range: 0x1- 0xFF

Description : The number of Data Frames after which NAC Frame has to be transmitted



## ERR\_TRG\_TIMEOUT



Figure 23: ERR\_TRG\_TIMEOUT

This function will cause Link-Start to fail by not allowing TRG\_UPR of the type specified to be sent in the Tx direction, thus not allowing the device or tester to enter the next phase.

### Parameters:

args:

Input type: Enumeration

Range: TRG\_UPR0, TRG\_UPR1, TRG\_UPR2 .

Description: Defines which TRG\_UPR are to be sent continuously for 100ms so that LinkStartup Time-out occurs

## ERR\_TRG\_TIMEOUT\_RX

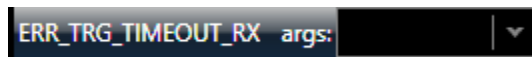


Figure 24: ERR\_TRG\_TIMEOUT\_RX

This function will cause Link-Start to fail by not allowing TRG\_UPR of the type specified to be sent in the Rx direction, thus not allowing the device or tester to enter the next phase.

### Parameters:

args:

Input type: Enumeration

Range: TRG\_UPR0, TRG\_UPR1, TRG\_UPR2 .

Description: Defines which TRG\_UPR are to be sent continuously for 100ms so that LinkStartup Time-out occurs





## ERR\_IN\_TCO



Figure 25: ERR\_IN\_TCO

This function generates an error in a TCO frame. Limited to removing the CRC or creating a corrupted CRC.

Parameters:

arg::

Input Type: Enumeration

Range: RM\_CRC, BAD\_CRC

Description: How to change the CRC to create the error condition

Length:

Input Type: Integer

Range: 0-65,535

Description: Length to change the reported packet payload to.

## ERR\_SKIP\_AFC



Figure 26: ERR\_SKIP\_AFC

This function indicates the number of scheduled AFC Frames that are not to be transmitted. Currently this function is not implemented.



## EnableTstDst



Figure 27: EnableTstDst

This function will enable the local traffic analyzer TstDst with the specified parameters. If Param1 (ErrorCheck) is set, TstDst will check the size and payload of the incoming Messages based on the value of Param2 (Increment) and Param3 (MessageSize). Otherwise, TstDst will accept incoming Messages without generating any error Messages.

### Parameters:

#### Param0:

Input type: Integer

Range: 0 to T\_NumCPorts - 1

Description: Defines the Cport index to target

#### Param1:

Input type: Boolean

Range: 0-1 (FALSE=0, TRUE=1)

Description: Defines the ErrorCheck value to use

#### Param2:

Input type: Integer

Range: 0 - 255

Description: Defines the Increment value to use

#### Param3:

Input type: Integer

Range: 0 – 65,535

Description: Defines the MessageSize value to use



## WaitForMessageCount



Figure 28: WaitForMessageCount

This function causes the link to wait for N number of messages from TstDst, or until the specified TimeOut expires.

### Parameters:

#### Parm0:

Input type: Integer

Range: 0 - 65,535

Description: Message count to wait before proceeding

#### Parm1:

Input type: Integer

Range: 0 - 65,535

Description: Timeout in mS to wait before proceeding if message count is not met.  
Defaults to 500 mS.



## Variables

### Global Variable

Global Variables can be used for Run Conditions or within a Test Case.

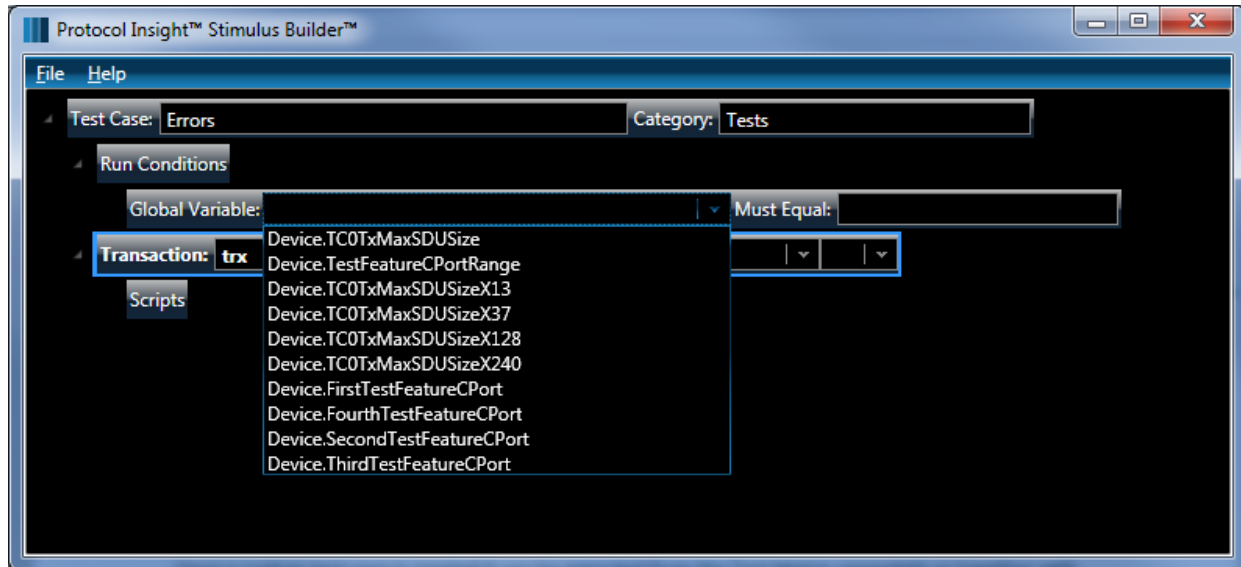


Figure 29: Global Variables

The following global variables are available:

- Device.TC0TxMxSDUSize
- Device.TestFeaturesCPortRange
- Device.TC0MaxSDUSizeX13
- Device. TC0TxMxSDUSizeX37
- Device. TC0TxMxSDUSizeX128
- Device. TC0TxMxSDUSizeX240
- Device.FirstTestFeatureCPort
- Device.SecondTestFeatureCPort
- Device.ThirdTestFeatureCPort
- Device.FourthTestFeatureCPort



## Payload

A payload contains the commands Test Executive transmits to the DUT.

Command:	Command	Gap:	0	Repeat Count:	1	Invert CRC:	FALSE
Sequence Number Increment:	0	SOF Error:	FALSE	SOF Replacement:	0		
L4Payload	Type: UpCounter	Length:	1				

Figure 30: A Payload

This function generates L4 data payload

### Parameters

- Command: This is a name that can be applied to the payload to describe its function
- Gap: The gap between two consecutive data frames
- Repeat Count: The number of times the message is to be repeated
- Invert CRC: Creates an error in the CRC
- Sequence Number Increment: Increments the sequence number by value entered
- SOF Error: Creates a Start of Frame error
- SOF Replacement: Replaces the SOF with value



## Appendix A: Syntax Checker Error Messages

The following test case construction rules are verified by the syntax checker for a custom test case:

1. The Test Case must have a name specified.
2. There must be at least 1 Transaction defined in the Stimulus.
3. The Transaction must have at least 1 payload or 1 script.



## Appendix B: Known Limitations

The following are the known limitations of Falcon UniPro Stimulus Builder:

- None

© Protocol Insight 2016, 2017, 2018

Specifications and descriptions in this document  
are subject to change without notice.  
© Protocol Insight, LLC 2016, 2017, 2018  
Revision 1.6.0, March 30, 2018