TELEDYNE LECROY
Everywhereyoulook™

USB4 Exerciser Manual

Manual Version 1.20

For USB Protocol Suite Software Version 8.50 and above

December 2020

Document Disclaimer

The information contained in this document has been carefully checked and is believed to be reliable. However, no responsibility can be assumed for inaccuracies that may not have been detected.

Teledyne LeCroy reserves the right to revise the information presented in this document without notice or penalty.

Trademarks and Servicemarks

CATC Trace, Voyager M4x™, Voyager ReadyLink, USB Protocol Suite, and *BusEngine* are trademarks of Teledyne LeCroy.

All other trademarks are property of their respective companies.

Copyright

Contents

1. Introduction

Integrated in Teledyne LeCroy's Voyager M4x test platform, the USB4 exerciser supports traffic generation, including both Host and Device emulation. The USB4 exerciser continues to evolve with each software release. Be sure to check for updated software and firmware before getting started with the Exerciser.

Important Licensing Note:

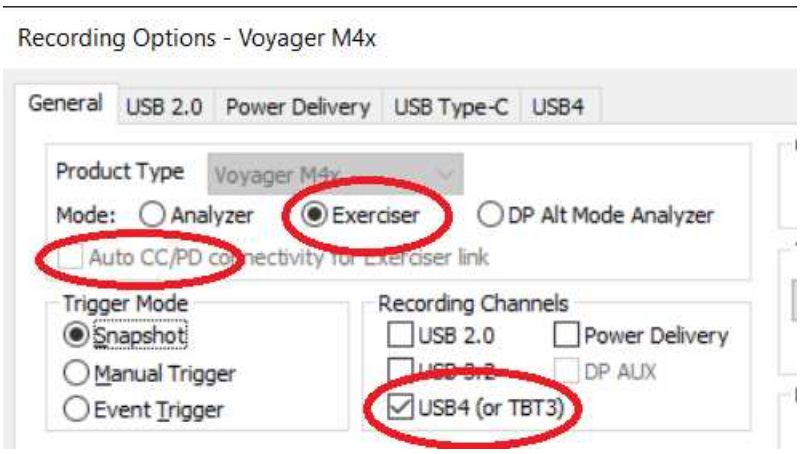
- Operating theUSB4 Exerciser requires that the USB4 Exerciser option is enabled on the M4x base unit, check unit license:

Getting Started:

- The port that labeled as "Exerciser" should be used to connect DUT to the USB4 Exerciser.

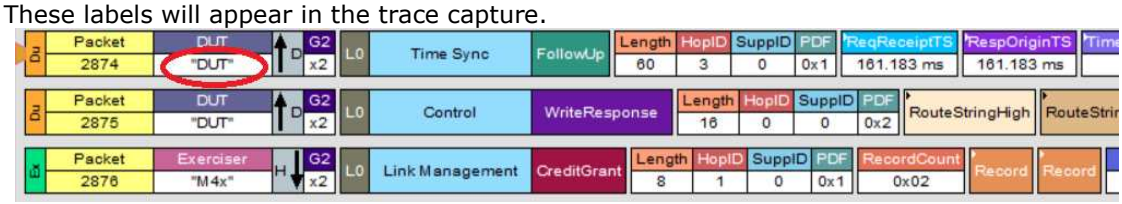
- The USB4 exerciser also requires specific cable orientation (Red LED when connected wrong side-up).

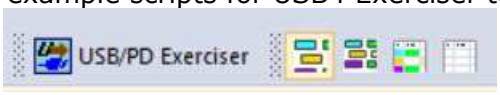
- To enable the USB4 Exerciser, set working mode to "Exerciser" in general page of "Recording options" and make sure "Auto CC/PD connectivity" is unchecked. Recording channel is selected to USB4.



- To set *devices port name*, use the General Tab under *"Recording Options"* to add "alias labels" for your DUTs.

These labels will appear in the trace capture.



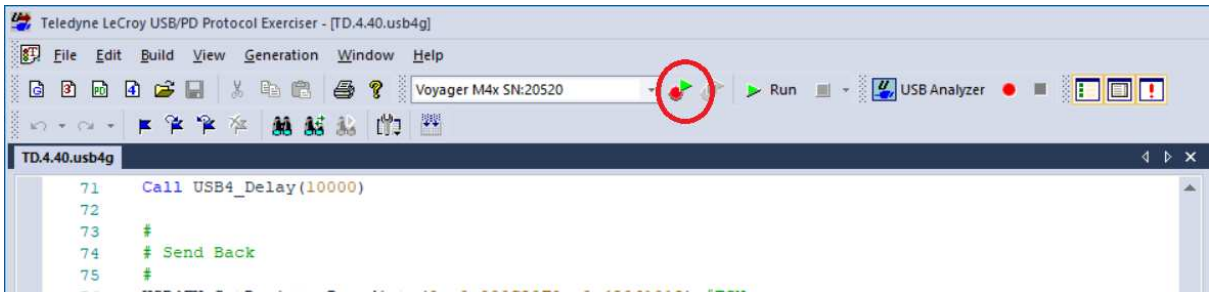
- Use the example scripts for USB4 Exerciser to begin testing. Open script editor GUI by clicking 

icon in main application toolbar and open one of following examples located at:

  C:\Users\Public\Documents\LeCroy\USB Protocol Suite\Examples\USB4 Exerciser

Example Script	Behaviour
Host_Emulator.usb4g	Voyager M4x is emulated host that enumerate DUT connected to exerciser port to bonding state
SB_Error_Injection.usb4g	Voyager M4x is emulated host that enumerate DUT connected to exerciser port to bonding state and then injects a SB packet with invalid CLSE

- To Run Sample Script – Connect DUT to Exerciser port; Click *Record*/Run exerciser icon in script editor toolbar



2. Scope of the Document

The document is intended as an extension to Voyager Exerciser language. It addresses only the new elements specific to USB4 Exerciser on top of the language described in "**Voyager Exerciser Language Manual**" (accessible through Menu -> Help -> Other Manuals).

3. Packet Templates

Name	Description
USB4VarContainer	A predefined template with a variable length field.
USB4DWordContainer	A predefined template with a DWORD field.
USB4TPHeader	
USB4RouteString	
USB4ReadWriteControlData	
USB4ReadRequest	
USB4ReadResponse	
USB4WriteRequest	
USB4WriteResponse	
USB4NotificationAck	
USB4HotPlugEvent	
USB4Notification	
USB4_LT_Transaction	
USB4_AT_Transaction	
USB4_Broadcast_RT_Transaction	
USB4_Addressed_RT_Transaction	
USB4RouterCfgSpace	
ROUTER_CS_n*	n : 0 to 26
USB4TMU_RouterCfgCap	
TMU_RTR_CS_n*	n : 0 to 25
USB4VSC_RouterCfgCap	
VSC_CS_0	
USB4VSEC_RouterCfgCap	
VSEC_CS_n*	n : 0 to 1
USB4BasicAdapterCfgSpace	
ADP_CS_n*	n : 0 to 8
USB4TMU_AdapterCfgCap	
TMU_ADP_CS_n*	n : 0 to 7
USB4LaneAdapterCfgCap	
LANE_ADP_CS_n	n : 0 to 2
USB4PortCapability	
PORT_CS_n*	n : 0 to 19
USB4USB3PortCapability	
ADP_USB3_CS_n*	n : 0 to 4
USB4DP_IN_Adapter	
IN_ADP_DP_CS_n*	n : 0 to 3
IN_DP_LOCAL_CAP	
IN_DP_REMOTE_CAP	
IN_DP_STATUS	
IN_DP_COMMON_CAP	
USB4DP_OUT_Adapter	
OUT_ADP_DP_CS_n*	n : 0 to 3
OUT_DP_LOCAL_CAP	
OUT_DP_REMOTE_CAP	
OUT_DP_STATUS	
OUT_DP_COMMON_CAP	
OUT_ADP_DP_CS_8	

USB4PCIe_Adapter	
ADP_PCIE_CS_0	

4. Global Settings

$USB4GlobalSettings is a global variable with following fields:

Field Name	Possible Values	Description
DataRole	USB4_DEVICE USB4_HOST (default)	
Protocol	USB4_PROTOCOL_TBT3 USB4_PROTOCOL_USB4 (default)	
PDWorkingRevision	USB4_PD_WORKING_REV_2 USB4_PD_WORKING_REV_3 (default)	
ForceDisconnectOnStop	USB4_FALSE USB4_TRUE (default)	Disconnect Type-C connection Forcefully when the Exerciser is stopped.
TxSSCAlwaysOn	USB4_FALSE USB4_TRUE (default)	Set Spread Spectrum Clock of the Exerciser Tx on/off.

Examples:

```
Set $USB4GlobalSettings.DataRole = USB4_DEVICE
Set $USB4GlobalSettings.Protocol  = USB4_PROTOCOL_TBT3
```

5. Sideband Registers

Sideband Registers fields can be accessed or changed through $USB4SBRegisters which is a global variable with following fields:

Field Name	Possible Values	Description
Vendor ID Register (0)		
VID_Low	0xFF (default)	
VID_High	0x05 (default)	
VID_Rsvd1	0 (default)	
VID_Rsvd2	0 (default)	
Product ID Register (1)		
PID_Low	0x04 (default)	
PID_High	0x00 (default)	
PID_Rsvd1	0 (default)	
PID_Rsvd2	0 (default)	
Opcode Register (8)		
OPC_0	0 (default)	
OPC_1	0 (default)	
OPC_2	0 (default)	
OPC_3	0 (default)	
MetaData Register (9)		
MD_Metadata	0 (default)	
Link Configuration Register (12)		
LC_EnablingDecisionLane0	USB4_FALSE (default) USB4_TRUE	
LC_EnablingDecisionLane1	USB4_FALSE (default) USB4_TRUE	
LC_Rsvd1	0 (default)	
LC_EnablingRequestLane0	USB4_FALSE USB4_TRUE (default)	
LC_EnablingRequestLane1	USB4_FALSE USB4_TRUE (default)	
LC_Rsvd2	0 (default)	
LC_BondingSupport	USB4_FALSE USB4_TRUE (default)	
LC_Gen3Support	USB4_FALSE USB4_TRUE (default)	
LC_RSFEC_RequestGen2	USB4_FALSE (default) USB4_TRUE	
LC_RSFEC_RequestGen3	USB4_FALSE USB4_TRUE (default)	
LC_USB4SidebandChannelSupport	USB4_FALSE USB4_TRUE (default)	
LC_TBT3CompatibleSpeedsSupport	USB4_FALSE (default) USB4_TRUE	
LC_Rsvd2	0 (default)	
TxFFE Register (13)		
TxFFE_RequestLane0	0 (default)	
TxFFE_RxLockedLane0	0 (default)	
TxFFE_RxActiveLane0	0 (default)	
TxFFE_ClockSwitchDoneLane0	0 (default)	
TxFFE_NewRequestLane0	0 (default)	
TxFFE_RequestLane1	0 (default)	
TxFFE_RxLockedLane1	0 (default)	
TxFFE_RxActiveLane1	0 (default)	
TxFFE_ClockSwitchDoneLane1	0 (default)	
TxFFE_NewRequestLane1	0 (default)	
TxFFE_SettingLane0	0 (default)	
TxFFE_Rsvd1	0 (default)	
TxFFE_RequestDoneLane0	0 (default)	
TxFFE_TxActiveLane0	0 (default)	

TxFFE_SettingLane1	0 (default)	
TxFFE_Rsvd2	0 (default)	
TxFFE_RequestDoneLane1	0 (default)	
TxFFE_TxActiveLane1	0 (default)	
Sideband Channel Version Register (15)		
CV_SubVersion	0 (default)	
CV_MinorVersion	0 (default)	
CV_MajorVersionLSB	0 (default)	
CV_MajorVersionMSB	0 (default)	
Data Register (18)		
DATA0	0 (default)	
DATA1	0 (default)	
DATA2	0 (default)	
DATA3	0 (default)	
DATA4	0 (default)	
DATA5	0 (default)	
DATA6	0 (default)	
DATA7	0 (default)	
DATA8	0 (default)	
DATA9	0 (default)	
DATA10	0 (default)	
DATA11	0 (default)	
DATA12	0 (default)	
DATA13	0 (default)	
DATA14	0 (default)	
DATA15	0 (default)	

Examples:

```
Set $USB4SBRegisters.LC_BondingSupport        = USB4_TRUE
Set $USB4SBRegisters.LC_Gen3Support           = USB4_FALSE
Set $USB4SBRegisters.LC_RSFEC_RequestGen2     = USB4_FALSE
Set $USB4SBRegisters.LC_RSFEC_RequestGen3     = USB4_FALSE
```

6. Command Settings

7. Commands

1. Set

It modifies a field in sent packet buffer, global settings and Exerciser's Sideband Registers. For setting a field in global settings and Sideband Registers, you need to use two predefined variables, `$USB4GlobalSettings` and `$USB4SBRegisters` respectively. But for the sent packet, you need to create your own custom variable inherited from USB4Packet. When such a variable is used as the first operand, it refers to the sent packet and if it is used as the second operand, it refers to the latest received packet.

Format:

```
Set [$Variable].[Field] = [numeric value]
Set [$Variable].[Field] = [$Variable].[Field]
```

Examples:

```
Set $USB4GlobalSettings.DataRole = USB4_DEVICE

or

Struct ControlPacketBase : USB4Packet
{
    : USB4TPHeader
    : USB4RouteString
    : USB4ReadWriteControlData
}

$sent_packet_header = ControlPacketBase


Set $sent_packet_header.AdapterNum = 3

# Copy HopId of the received packet to the sent packet
Set $sent_packet_header.HopId = $sent_packet_header.HopId

or

# Define a Control packet with one DWORD payload. It represents a Request command to read/write the third Register of the Lane Adapter Configuration space.
$sent_packet = ControlPacketBase
{
    : LANE_ADP_CS_2
}

Set $sent_packet.CS_2_LogicalLayerErr = 0x2
```

2. USB4_StartRouter

Starts a Router based on the specified Protocol and Data Role in $USB4GlobalSettings. It adds some default configuration spaces and handles PowerDelivery Discovery and Entry flow, sideband Lane initialization and Router initialization until after lane bonding.

Format:

```
USB4_StartRouter()
```

Example:

```
Call USB4_StartRouter()
```

3. USB4_DisconnectRouter

Exits PowerDelivery Alt mode and disconnect the Type-C connection.

Format:

```
USB4_DisconnectRouter()
```

Example:

```
Call USB4_DisconnectRouter()
```

4. USB4_Pause

It causes to pause execution of the Main function. It will be resumed upon calling ResumeUSB4Generation() API.

The Tag parameter give an ability to the WaitForUSB4GenerationPauseTag() API to differentiate different pause commands used in a script.

Format:

```
USB4_Pause( tag )
```

Examples:

```
Call USB4_Pause( 20 )
```

5. USB4_Delay

It adds a delay before running the next command in the script.

Format:

```
USB4_Delay( time_micro_sec )
```

Example:

```
Call USB4_Delay( 1000 )
```

6. USB4_SendSBTransaction

It Sends a Transaction on sideband channel. The parameter can be a variable of the following types:

Format:

```
USB4_SendSBTransaction( USB4SendSBTransactionSettings $settings, $transsaction_struct )
```

Parameters:

- USB4SendSBTransactionSettings:

Field Name	Possible Values	Description
AutoCLSE	USB4_FALSE USB4_TRUE (default)	Used for LT Transactions only. Set it to false if you want to have error injection on CLSE
AutoCRC	USB4_FALSE USB4_TRUE (default)	Used for AT, Broadcast RT and Addressed RT Transactions. Set to false for injecting CRC error.

- $transsaction_struct: A variable of the following templates to be sent on Sideband channel:
  - USB4_LT_Transaction
  - USB4_AT_Transaction
  - USB4_Broadcast_RT_Transaction
  - USB4_Addressed_RT_Transaction

Example:

```
$settings = USB4SendSBTransactionSettings
$transsaction_struct = USB4_LT_Transaction
{
        LSELane = 1
        CLSE = 0x00
}
Call USB4_SendSBTransaction( $settings, $transsaction_struct )
```

7. USB4_SendReadRequest

It sends a Control packet Read Request.

Format:

```
USB4_SendReadRequest( USB4SendCtrlReadWriteSettings $settings )
```

Parameter:

- USB4SendCtrlReadWriteSettings:

Field Name	Possible Values	Description
Lane	USB4_LANE_0 (default) USB4_LANE_1 USB4_LANE_BONDED	
RouteStringHigh	0 (default)	
CM	0 (default) 1	
RouteStringLow	0 (default)	
Address	0 (default)	If CapabilityID is zero, it is an absolute address otherwise it is an offset within selected capability by CapabilityID.
DataSize	0 (default)	
AdapterNum	0 (default)	
ConfigurationSpace	USB4_PATH_CFG USB4_ADAPTER_CFG USB4_ROUTER_CFG (default) USB4_COUNTERS_CFG	
SequenceNumber	0 (default)	Ignored if AutoSequenceNumber is true.
AutoSequenceNumber	USB4_FALSE USB4_TRUE (default)	If true, SequenceNumber is handled internally.
CapabilityID	0 (default) USB4_CFG_CAP_TMU USB4_CFG_CAP_LANE USB4_CFG_CAP_USB4 USB4_CFG_CAP_PCIE USB4_CFG_CAP_DP_IN USB4_CFG_CAP_DP_OUT USB4_CFG_CAP_USB3 USB4_CFG_CAP_VS	Refer to Address field.
WaitForResponse	USB4_FALSE USB4_TRUE (default)	If true, command waits to receive the response.
HECErrorMask	0x00	If it is not zero, the corresponding bit of each one in this mask will be flipped in HEC field to inject HEC error.
CRC32ErrorMask	0x00000000	If it is not zero, the corresponding bit of each one in this mask will be flipped in CRC32 field to inject CRC error.

Example:

```
$settings = USB4SendCtrlReadWriteSettings
{
        Lane = USB4_LANE_0

        RouteStringHigh = 0x00000000
        RouteStringLow  = 0x00000001

        Address = 0x02 # offset within TMU capability
        CapabilityID = USB4_CFG_CAP_TMU

        ConfigurationSpace = USB4_ROUTER_CFG
}
Call USB4_SendReadRequest( $settings )
```

8. USB4_SendWriteResponse

It sends a Control packet write Response.

Format:

```
USB4_SendWriteResponse( USB4SendCtrlReadWriteSettings $settings )
```

Parameter:

- USB4SendCtrlReadWriteSettings:

Field Name	Possible Values	Description
Lane	USB4_LANE_0 (default) USB4_LANE_1 USB4_LANE_BONDED	
RouteStringHigh	0 (default)	Ignored if AutoRouteString is true.
CM	0 (default) 1	Ignored if AutoRouteString is true.
RouteStringLow	0 (default)	Ignored if AutoRouteString is true.
Address	0 (default)	
DataSize	0 (default)	
AdapterNum	0 (default)	
ConfigurationSpace	USB4_PATH_CFG USB4_ADAPTER_CFG USB4_ROUTER_CFG (default) USB4_COUNTERS_CFG	
SequenceNumber	0 (default)	Ignored if AutoSequenceNumber is true.
AutoSequenceNumber	USB4_FALSE USB4_TRUE (default)	If true, SequenceNumber is copied from the latest received Control packet internally.
AutoRouteString	USB4_FALSE USB4_TRUE (default)	If true, RouteString is copied from the latest received Control packet internally.
HECErrorMask	0x00	If it is not zero, the corresponding bit of each one in this mask will be flipped in HEC field to inject HEC error.
CRC32ErrorMask	0x00000000	If it is not zero, the corresponding bit of each one in this mask will be flipped in CRC32 field to inject CRC error.

Example:

```
$settings = USB4SendCtrlReadWriteSettings
{
        Lane = USB4_LANE_1

        RouteStringHigh = 0x00000000
        RouteStringLow  = 0x00000001
}
Call USB4_SendWriteResponse ( $settings )
```

9. USB4_SendReadResponse

It sends a Control packet Read Response.

Format:

```
USB4_SendReadResponse( USB4SendCtrlReadWriteSettings $settings, $data )
```

Parameter:

- USB4SendCtrlReadWriteSettings:

Field Name	Possible Values	Description
Lane	USB4_LANE_0 (default) USB4_LANE_1 USB4_LANE_BONDED	
RouteStringHigh	0 (default)	Ignored if AutoRouteString is true.
CM	0 (default) 1	Ignored if AutoRouteString is true.
RouteStringLow	0 (default)	Ignored if AutoRouteString is true.
Address	0 (default)	
DataSize	0 (default)	
AdapterNum	0 (default)	
AutoAdapterNum	USB4_FALSE USB4_TRUE (default)	
ConfigurationSpace	USB4_PATH_CFG USB4_ADAPTER_CFG USB4_ROUTER_CFG (default) USB4_COUNTERS_CFG	
SequenceNumber	0 (default)	Ignored if AutoSequenceNumber is true.
AutoSequenceNumber	USB4_FALSE USB4_TRUE (default)	If true, SequenceNumber is copied from the latest received Control packet internally.
AutoRouteString	USB4_FALSE USB4_TRUE (default)	If true, RouteString is copied from the latest received Control packet internally.
HECErrorMask	0x00	If it is not zero, the corresponding bit of each one in this mask will be flipped in HEC field to inject HEC error.
CRC32ErrorMask	0x00000000	If it is not zero, the corresponding bit of each one in this mask will be flipped in CRC32 field to inject CRC error.

- $data: Data payload of the Read Response

Example:

```
$data = USB4VarContainer
{
        Data = { 00 11 22 33 }
}
$settings = USB4SendCtrlReadWriteSettings
{
        Lane = USB4_LANE_0

        RouteStringHigh = 0x00000000
        RouteStringLow  = 0x00000001

        Address = 0
        DataSize = 1
        AdapterNum = 1
        ConfigurationSpace = USB4_ADAPTER_CFG
}
Call USB4_SendReadResponse( $settings, $data )
```

10. USB4_SendWriteRequest

It sends a Control packet Write Request. If ImmediatePayload field in the settings is true, the second parameter is used as the Write Data Payload otherwise the provided data in the send buffer is used.

Format:

USB4_SendWriteRequest(USB4SendCtrlReadWriteSettings $settings, $data)

Parameter:

- USB4SendCtrlReadWriteSettings:

Field Name	Possible Values	Description
Lane	USB4_LANE_0 (default) USB4_LANE_1 USB4_LANE_BONDED	
RouteStringHigh	0 (default)	Ignored if AutoRouteString is true.
CM	0 (default) 1	Ignored if AutoRouteString is true.
RouteStringLow	0 (default)	Ignored if AutoRouteString is true.
Address	0 (default)	If CapabilityID is zero, it is an absolute address otherwise it is an offset within selected capability by CapabilityID.
DataSize	0 (default)	
AdapterNum	0 (default)	
AutoAdapterNum	USB4_FALSE USB4_TRUE (default)	
ConfigurationSpace	USB4_PATH_CFG USB4_ADAPTER_CFG USB4_ROUTER_CFG (default) USB4_COUNTERS_CFG	
SequenceNumber	0 (default)	Ignored if AutoSequenceNumber is true.
AutoSequenceNumber	USB4_FALSE USB4_TRUE (default)	If true, SequenceNumber is copied from the latest received Control packet internally.
AutoRouteString	USB4_FALSE USB4_TRUE (default)	If true, RouteString is copied from the latest received Control packet internally.
ImmediatePayload	USB4_FALSE USB4_TRUE (default)	If true, the passed data to the command is used as Read Response payload otherwise the data in the send buffer is used.
CapabilityID	0 (default) USB4_CFG_CAP_TMU USB4_CFG_CAP_LANE USB4_CFG_CAP_USB4 USB4_CFG_CAP_PCIE USB4_CFG_CAP_DP_IN USB4_CFG_CAP_DP_OUT USB4_CFG_CAP_USB3 USB4_CFG_CAP_VS	Refer to Address field.
WaitForResponse	USB4_FALSE USB4_TRUE (default)	If true, command waits to receive the response.
HECErrorMask	0x00	If it is not zero, the corresponding bit of each one in this mask will be flipped in HEC field to inject HEC error.
CRC32ErrorMask	0x00000000	If it is not zero, the corresponding bit of each one in this mask will be flipped in CRC32 field to inject CRC error.

- $data: Data payload of the Write Request

Example:

```
$data = USB4VarContainer
{
        Data = { 00 11 22 33 }
}
$settings = USB4SendCtrlReadWriteSettings
{
        Lane = USB4_LANE_0

        RouteStringHigh = 0x00000000
        RouteStringLow  = 0x00000001

        Address = 0
        DataSize = 1
        AdapterNum = 1
        ConfigurationSpace = USB4_ADAPTER_CFG
}
Call USB4_SendWriteRequest( $settings, $data )
```

11. USB4_ResetRouterCfg

It clears all added structures to Router Configuration space.

Format:

```
USB4_ResetRouterCfg( router_id )
```

12. USB4_AddRouterCfg

It appends a structure to the Router Configuration space. Note that you need to set the proper NextCapPointer manually.

Format:

```
USB4_AddRouterCfg( router_id, USB4TemplateBase $cfg_struct )
```

Example:

```
struct_size_dw = pkt_size( USB4RouterCfgSpace ) / 4
$router_cfg_struct = USB4RouterCfgSpace
{
        CS_0_VendorID  = 0x05FF
        CS_0_ProductID = 0x0004

        CS_1_NextCapPointer   = struct_size_dw
        CS_1_MaxAdapter       = 5
        CS_1_UpstreamAdapter = 1
        CS_1_RevisionNumber  = 0x6
}
Call USB4_AddRouterCfg( 0, $router_cfg_struct )
```

13. USB4_AddAdapterCfg

It appends a structure to Configuration space of the specified Adapter. Note that you need to set the proper NextCapPointer manually.

Format:

```
USB4_AddRouterCfg( router_id, adapter_num, USB4TemplateBase $cfg_struct )
```

Example:

```
struct_size_dw = pkt_size( USB4BasicAdapterCfgSpace ) / 4
$adapter_cfg_struct_baseic = USB4BasicAdapterCfgSpace
{
        CS_1_NextCapPointer  = struct_size_dw

        CS_2_AdpTypeSubtype  = USB4_ADP_TYPE_LANE
        CS_2_AdpTypeVersion  = USB4_ADP_VER_LANE
        CS_2_AdpTypeProtocol = USB4_ADP_PRO_LANE

        CS_3_AdapterNumber = 1

        CS_5_MaxInputHopID  = 19
        CS_5_MaxOutputHopID = 19
}
USB4_AddRouterCfg( 0, 1, $adapter_cfg_struct_baseic )
```

14. USB4_ResetAdapterCfg

It clears all added structures to the specified Adapter Configuration space.

Format:

```
USB4_ResetRouterCfg( router_id, adapter_num )
```

15. USB4_SetDefaultCfgSpacesHost

16. USB4_SetDefaultCfgSpacesDevice

Adds some default Configuration Spaces based on $USB4GlobalSettings.DataRole.

To add your own Configuration Spaces, you need to use USB4_AddRouterCfg() and USB4_AddAdapterCfg() commands.

Format:

```
USB4_SetDefaultCfgSpacesHost()
USB4_SetDefaultCfgSpacesDevice()
```

Example:

```
Call USB4_SetDefaultCfgSpacesHost()
```

17. USB4_EnterTBT3AltModeHost
18. USB4_EnterTBT3AltModeDevice

Handle Type-C connectivity and Power Delivery TBT3 discovery and entry process.

Format:

USB4_EnterTBT3AltModeHost()
USB4_EnterTBT3AltModeDevice()

Example:

Call USB4_EnterTBT3AltModeHost()

19. USB4_EnterUSB4AltModeHost
20. USB4_EnterUSB4AltModeDevice

Handle Type-C connectivity and Power Delivery USB4 discovery and entry process.

Format:

USB4_EnterUSB4AltModeHost()
USB4_EnterUSB4AltModeDevice()

Example:

Call USB4_EnterUSB4AltModeHost()

21. USB4_StartCM

It starts Connection Manager's tasks including Router enumeration, Adapters enumeration and Lane Bonding.

Format:

USB4_StartCM()

Examples:

Call USB4_StartCM()

22. USB4_CopyRcvdPacketToSendBuffer()

This command copies the received Control packet to send buffer. It is used when USB4 script wants to change Configuration Spaces of a Router under test. For that purpose, a Read request is initiated to read the specified address. Then received packet is copied to the send buffer by this command and required change is applied to it. At the last, a Write request is issued to write the data to the Router.

Format:

```
USB4_CopyRcvdPacketToSendBuffer()
```

Example:

```
local $send_settings = USB4SendCtrlReadWriteSettings
{
   Lane = USB4_LANE_BONDED

   RouteStringHigh    = 0x00000000
   RouteStringLow     = 0x00000003

   Address          = 0x38
   DataSize         = 1
   AdapterNum       = 1
   ConfigurationSpace = USB4_ADAPTER_CFG
}
Call USB4_SendReadRequest($send_settings)

Call USB4_CopyRcvdPacketToSendBuffer()

Struct ReadLaneAdp_CS_2 : USB4ReadWriteControlData
{
   : LANE_ADP_CS_2

}
local $sent_packet = ReadLaneAdp_CS_2

Set $sent_packet.CS_2_LogicalLayerErrEnable = 1


$send_settings
{
   ImmediatePayload = USB4_FALSE
}
Call USB4_SendWriteRequest( $send_settings )
```

23. USB4_WaitForCL0

It waits until the specified lanes transition to CL0 state. Possible parameters are:

USB4_LANE_0

USB4_LANE_1

USB4_LANE_BOTH

Format:

```
USB4_WaitForCL0( lane )
```

Examples:

```
Call USB4_WaitForCL0( USB4_LANE_BOTH )
```

24. USB4_SetDROMSpace

It sets DROM space.

Format:

```
USB4_SetDROMSpace( $drom )
```

Example:

```
$drom = USB4VarContainer
{
    Data = { 86
             00 BC FD 45 76 1F 86 80
             08 DA 31 2A
             01
             19 00
             86 80
             34 12
             03
             02
             08 81 80 02 00 00 00 00
             08 82 90 01 00 00 00 00 }
}
Call USB4_SetDROMSpace ( $drom )
```

25. USB4_SendHotPlugEvent

It sends a Hot Plug Event.

Format:

```
USB4_SendHotPlugEvent( USB4SendHotPlugSettings $settings )
```

Parameter:

- USB4SendHotPlugSettings:

Field Name	Possible Values	Description
Lane	USB4_LANE_0 (default) USB4_LANE_1 USB4_LANE_BONDED	
RouteStringHigh	0 (default)	Ignored if AutoRouteString is true.
CM	0 (default) 1	Ignored if AutoRouteString is true.
RouteStringLow	0 (default)	Ignored if AutoRouteString is true.
AdapterNum	0 (default)	
UPG	0 (default)	
WaitForResponse	USB4_FALSE USB4_TRUE (default)	If true, command waits to receive the response.

Example:

```
$settings = USB4SendHotPlugSettings
{
        Lane = USB4_LANE_0

        RouteStringHigh = 0x00000000
        RouteStringLow  = 0x00000001

        AdapterNum = 1
        UPG = 1
}
Call USB4_SendHotPlugEvent( $settings )
```

26. USB4_SendNotification

It sends a Notification packet.

Format:

```
USB4_SendNotification( USB4SendNotificationSettings $settings )
```

Parameter:

- USB4SendNotificationSettings:

Field Name	Possible Values	Description
Lane	USB4_LANE_0 (default) USB4_LANE_1 USB4_LANE_BONDED	
RouteStringHigh	0 (default)	Ignored if AutoRouteString is true.
CM	0 (default) 1	Ignored if AutoRouteString is true.
RouteStringLow	0 (default)	Ignored if AutoRouteString is true.
EventCode	USB4_ERR_CONN (default) USB4_ERR_LINK USB4_ERR_ADDR USB4_ERR_ADP USB4_ERR_ENUM USB4_ERR_NUA USB4_ERR_LEN USB4_ERR_HEC USB4_ERR_FC USB4_ERR_PLUG USB4_ERR_LOCK USB4_HP_ACK	
AdapterNum	0 (default)	
PG	USB4_NOTIFICATION (default) USB4_HOT_PLUG_EVENT USB4_HOT_UNPLUG_EVENT	
WaitForResponse	USB4_FALSE USB4_TRUE (default)	If true, command waits to receive the response.
HECErrorMask	0x00	If it is not zero, the corresponding bit of each one in this mask will be flipped in HEC field to inject HEC error.
CRC32ErrorMask	0x00000000	If it is not zero, the corresponding bit of each one in this mask will be flipped in CRC32 field to inject CRC error.

Example:

```
$settings = USB4SendNotificationSettings
{
        Lane = USB4_LANE_0

        RouteStringHigh = 0x00000000
        RouteStringLow  = 0x00000001

        EventCode = USB4_ERR_ADDR
        AdapterNum = 1
        PG = USB4_NOTIFICATION
}
Call USB4_SendNotification ( $settings )
```

27. USB4_SendInterDomainRequest

It sends an Inter-Domain request.

Format:

```
USB4_SendInterDomainRequest ( USB4SendCtrlInterDomainSettings $settings )
```

Parameter:

- USB4SendCtrlInterDomainSettings:

Field Name	Possible Values	Description
Lane	USB4_LANE_0 (default) USB4_LANE_1 USB4_LANE_BONDED	
RouteStringHigh	0 (default)	Ignored if AutoRouteString is true.
CM	0 (default) 1	Ignored if AutoRouteString is true.
RouteStringLow	0 (default)	Ignored if AutoRouteString is true.
SN	0 (default)	
DiscoveryProtocol	0 (default)	
PacketType	0 (default)	
Data		InterDomain Data
WaitForResponse	USB4_FALSE USB4_TRUE (default)	If true, command waits to receive the response.
HECErrorMask	0x00	If it is not zero, the corresponding bit of each one in this mask will be flipped in HEC field to inject HEC error.
CRC32ErrorMask	0x00000000	If it is not zero, the corresponding bit of each one in this mask will be flipped in CRC32 field to inject CRC error.

Example:

```
$settings = USB4SendCtrlInterDomainSettings
{
        Lane = USB4_LANE_0

        RouteStringHigh = 0x00000000
        RouteStringLow  = 0x00000001

        DiscoveryProtocol = { B6 38 D7 0E 42 FF 40 BB 97 C2 90 E2 C0 B2 FF 07 }
        PacketType = 3
}
Call USB4_SendInterDomainRequest ( $settings )
```

28. USB4_SendTunneledPacket

It sends a Tunneled packet.

Format:

USB4_SendTunneledPacket (USB4SendTunneledPacketSettings$settings, $Payload)

Parameter:

- USB4SendTunneledPacketSettings:

Field Name	Possible Values	Description
Lane	USB4_LANE_0 (default) USB4_LANE_1 USB4_LANE_BONDED	
HopId	0 (default)	
PDF	0 (default)	

Example:

```
$payload = USB4VarContainer
{
        Data = { 00 11 22 33 }
}
$settings = USB4SendTunneledPacketSettings
{
        Lane = USB4_LANE_0

        HopId = 0x00000000
        PDF = 0x0000000
}
Call USB4_SendTunneledPacket ( $settings, $payload )
```

29. USB4_SendPathCreditSync

It sends a Path Credit Sync packet.

Format:

```
USB4_SendPathCreditSync ( USB4SendPathCreditSyncSettings $settings )
```

Parameter:

- USB4SendPathCreditSyncSettings:

Field Name	Possible Values	Description
Lane	USB4_LANE_0 (default) USB4_LANE_1 USB4_LANE_BONDED	
HopID	0 (default)	
PCC	0 (default)	
HECErrorMask	0x00	If it is not zero, the corresponding bit of each one in this mask will be flipped in HEC field to inject HEC error.
ECCErrorMask	0x00	If it is not zero, the corresponding bit of each one in this mask will be flipped in ECC field to inject ECC error.

Example:

```
$settings = USB4SendPathCreditSyncSettings
{
        Lane = USB4_LANE_0

        HopID = 8
        PCC = 1
}
Call USB4_SendPathCreditSync ( $settings )
```

30. USB4_InjectHSErrors

To inject some errors in high-speed traffic.

Format:

```
USB4_InjectHSErrors( USB4HSErrorInjectionSettings $settings )
```

Parameter:

- USB4HSErrorInjectionSettings:

Field Name	Possible Values	Description
BackToBackUnknownOS	USB4_FALSE (default) USB4_TRUE	Send back-to-back Ordered Sets with unknown contents
BackToBackSCRUncorrectable	USB4_FALSE (default) USB4_TRUE	Send back-to-back Ordered Sets with an uncorrectable error in the SCR field
SendTS1TillDisconnect	USB4_FALSE (default) USB4_TRUE	Keep sending TS1 to prevent transition to CL0 state
BondedRandomFECError	USB4_FALSE (default) USB4_TRUE	Inject randomly spaced errors so that either one or two errors fall within an RS-FEC block
LateBroadcastRT	USB4_FALSE (default) USB4_TRUE	Start sending Broadcast RT Transaction at the latest time
EarlyBroadcastRT	USB4_FALSE (default) USB4_TRUE	Start sending Broadcast RT Transaction at the earliest time
TS1Delay	0 (default)	Delay sending TS1 for the specified time
TS2Delay	0 (default)	Delay sending TS2 for the specified time
TS1SymbolsMask	0x0000000000000000 (default)	Specified which symbol(s) of send TS1 OS must be replaced with the corresponding symbol(s) in the TS1Symbols.
TS1Symbols		To override sent TS1 symbol.
TS2SymbolsMask	0x0000000000000000 (default)	Specified which symbol(s) of send TS2 OS must be replaced with the corresponding symbol(s) in the TS2Symbols.
TS2Symbols		To override sent TS2 symbol.

Example:

```
$error_injection = USB4HSErrorInjectionSettings
{
    BackToBackUnknownOS        = USB4_FALSE
    BackToBackSCRUncorrectable = USB4_TRUE
}
Call USB4_InjectHSErrors( $error_injection )
```

31. USB4_ClearInjectedHSErrors

It clears the injected errors by the USB4_InjectHSErrors() command.

Format:

```
USB4_ClearInjectedHSErrors( USB4HSErrorInjectionSettings $settings )
```

Parameter:

- USB4HSErrorInjectionSettings:

Field Name	Possible Values	Description
SendTS1TillDisconnect	USB4_FALSE (default) USB4_TRUE	Set true to stop injecting error
TS1Delay	0 (default)	Set nonzero to stop injecting error
TS2Delay	0 (default)	Set nonzero to stop injecting error

Example:

```
$clear_error = USB4HSErrorInjectionSettings
{
    SendTS1TillDisconnect = USB4_ TRUE
    TS1Delay = USB4_TRUE
    TS2Delay = USB4_FALSE
}
Call USB4_ClearInjectedHSErrors( $clear_error )
```

32. USB4_LoopbackRcvdPacket

To Loop Back the latest received control packet.

Format:

```
USB4_LoopbackRcvdPacket ( USB4LoopBackSettings $settings )
```

Parameter:

- USB4LoopBackSettings:

Field Name	Possible Values	Description
HECErrorMask	0x00	If it is not zero, the corresponding bit of each one in this mask will be flipped in HEC field to inject HEC error.
CRC32ErrorMask	0x00000000	If it is not zero, the corresponding bit of each one in this mask will be flipped in CRC32 field to inject CRC error.

Example:

```
$settings = USB4LoopBackSettings
{
    HECErrorMask = 0x01
}
Call USB4_LoopbackRcvdPacket( $settings )
```

33. USB4_WaitForPacketEx

It waits to receive a packet matched with input filter.

Format:

USB4_WaitForPacketEx(timeout_us, $filter_mask, $filter_match, USB4WaitForPacketFilterSettings $settings)

Parameter:

- USB4WaitForPacketFilterSettings:

Field Name	Possible Values	Description
FilterAutoResponse	USB4_FALSE USB4_TRUE (default)	If true, the proper response is send automatically before sending the received packet to USB4_WaitForPacket(). Set to false if you want to send a user defined response for the matched received packet.

Example:

```
$mask_value = USB4TPHeader
{
    HopId = 0xFFFF
}
$match_value = USB4TPHeader
{
    HopId = 8
}
$wait_settings = USB4WaitForPacketFilterSettings

Call USB4_WaitForPacketEx( 100000, $mask_value, $match_value, $wait_settings )
```

34. USB4_AddWaitForPacketFilter

To add a mask/match for USB4_WaitForPacket(). Currently, only one filter can be added. It is very important to call USB4_RemoveWaitForPacketFilters() after you finished with calling USB4_WaitForPacket().

Format:

USB4_AddWaitForPacketFilter($filter_mask, $filter_match, USB4WaitForPacketFilterSettings $settings)

Parameter:

- USB4WaitForPacketFilterSettings:

Field Name	Possible Values	Description
FilterAutoResponse	USB4_FALSE USB4_TRUE (default)	If true, the proper response is send automatically before sending the received packet to USB4_WaitForPacket(). Set to false if you want to send a user defined response for the matched received packet.

Example:

```
$mask_value = USB4TPHeader
{
    HopId = 0xFFFF
}
$match_value = USB4TPHeader
{
    HopId = 8
}
$wait_settings = USB4WaitForPacketFilterSettings

Call USB4_AddWaitForPacketFilter( $mask_value, $match_value, $wait_settings )
Call USB4_WaitForPacket( 100000 ) # wait for 100ms
Call USB4_RemoveWaitForPacketFilters()
```

35. USB4_WaitForPacket

It waits to receive a packet matched with added filters by USB4_AddWaitForPacketFilter().

Format:

```
USB4_WaitForPacket( timeout_us )
```

Examples:

```
$mask_value = USB4TPHeader
{
    HopId = 0xFFFF
}
$match_value = USB4TPHeader
{
    HopId = 8
}
$wait_settings = USB4WaitForPacketFilterSettings

Call USB4_AddWaitForPacketFilter( $mask_value, $match_value, $wait_settings )
Call USB4_WaitForPacket( 100000 ) # wait for 100ms
Call USB4_RemoveWaitForPacketFilters()
```

36. USB4_RemoveWaitForPacketFilters

It removes added filter by USB4_AddWaitForPacketFilter(). It is very important to call this function when you finished with calling USB4_WaitForPacket().

Format:

```
USB4_RemoveWaitForPacketFilters()
```

Examples:

```
Call USB4_RemoveWaitForPacketFilters()
```