



---

Protocol Solutions Group 3385 Scott Blvd., Santa Clara, CA 95054 Tel: +1/408.727.6600 Fax: +1/408.727.6622

# *PETrainer*

## Scripting Language

## Reference Manual

**Manual Version 1.30  
For PETracer Software Version 4.30**

18 February, 2005

## 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.

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

## Trademarks and Servicemarks

*CATC, PETrainer EML, PETrainer ML, PETracer EML, PETracer ML, PETracer, BusEngine* are trademarks of LeCroy.

*Microsoft, Windows, Windows 2000, and Windows XP* are registered trademarks of Microsoft Inc.

All other trademarks are property of their respective companies.

## Copyright

Copyright © 2005, LeCroy; All Rights Reserved.

This document may be printed and reproduced without additional permission, but all copies should contain this copyright notice.

## Version

This is version 1.30 the PETrainer Scripting Language Reference Manual. This manual applies to PETracer software version 4.30 and higher.

# Contents

<b>1</b>	<b>Introduction.....</b>	<b>1</b>
	1.1 Software Version .....	1
	1.2 Support Resources .....	1
<b>2</b>	<b>Syntax.....</b>	<b>2</b>
<b>3</b>	<b>Command List.....</b>	<b>2</b>
<b>4</b>	<b>Packet command.....</b>	<b>3</b>
	4.1 Packet = TLP.....	3
	4.2 Packet = DLLP.....	7
	4.3 Packet = OrderedSet .....	9
	4.4 Packet = Raw .....	9
	4.5 Packet = <TemplateName> .....	10
<b>5</b>	<b>Idle command.....</b>	<b>11</b>
<b>6</b>	<b>Link command .....</b>	<b>11</b>
	6.1 Link = L0 .....	11
	6.2 Link = L1 .....	11
	6.3 Link = L0s.....	11
	6.4 Link = Disabled.....	11
	6.5 Link = HotReset.....	11
	6.6 Link = Recovery .....	12
	6.7 Link = Detect .....	12
	6.8 Link = LTSSMOff.....	12
	6.9 Link = InitFC .....	12
<b>7</b>	<b>Config command.....</b>	<b>12</b>
	7.1 Config = General.....	12
	7.2 Config = FCTx.....	13
	7.3 Config = FCRx.....	13
	7.4 Config = TLP .....	13
	7.5 Config = AckNak.....	14
	7.6 Config = Transactions.....	14
	7.7 Config = Link.....	14
<b>8</b>	<b>Wait command.....</b>	<b>14</b>
	8.1 Wait = TLP .....	15
	8.2 Wait = DLLP.....	15
	8.3 Wait = Error .....	15
	8.4 Wait = Link .....	16
	8.5 Wait = BOB .....	16
	8.6 Wait = Payload.....	16

8.7	Wait = User .....	17
8.8	Additional “Wait” modifiers.....	17
<b>9</b>	<b>Template command .....</b>	<b>17</b>
<b>10</b>	<b>Loop command.....</b>	<b>18</b>
10.1	Loop=Begin .....	18
10.2	Loop=End .....	18
<b>11</b>	<b>Repeat command.....</b>	<b>18</b>
11.1	The Counter Parameter .....	19
11.2	Repeat=Begin.....	19
11.3	Repeat=End.....	19
<b>12</b>	<b>Branch command.....</b>	<b>20</b>
12.1	Branch = <condition> .....	20
12.2	Branch = Disable.....	20
<b>13</b>	<b>Proc command.....</b>	<b>21</b>
13.1	Proc = Begin .....	21
13.2	Proc = End.....	21
<b>14</b>	<b>Include command.....</b>	<b>21</b>

# 1 Introduction

This manual describes the scripting language used to create traffic generation files for PETrainer.

## 1.1 Software Version

This document is for: **PETracer software 4.30**

## 1.2 Support Resources

As new functionalities are added, not all of them are supported by older versions of the PETracer software. For newer releases of the analyzer's software, please refer to LeCroy's web site:

<http://www.lecroy.com/>

## 2 Syntax

PCI Express Generator Script file consists of the statements that have the following format:

```
COMMAND = MODIFIER
{
    PARAM1 = VALUE1
    ...
    PARAMn = VALUEn
}
```

All literals are not case sensitive.

All default values are zeros unless mentioned.

Integer literals represent numeric values with no fractions or decimal points.

Hexadecimal, decimal, and binary notation are supported:

Hexadecimal numbers must be preceded by 0x: 0x2A, 0x54, 0xFFFFFFFF01

Decimal numbers are written as usual: 24, 1256, 2

Binary numbers are denoted with 0b: 0b01101100, 0b01, 0b100000

It is possible to use expressions - for example, (0x239 - 239). See page 19 for more examples.

String literals are surrounded by double quotes.

Single-line comments are supported and should be preceded semicolon (“;”).

## 3 Command List

COMMAND	MODIFIERS	Comment
Packet	TLP DLLP OrderedSet Raw <TemplateName>	Sends packet
Idle	<# of ns>	Sends idle symbols (D0.0)
Link	L0 L1 L0s HotReset Disabled Recovery Detect LTSSMOff InitFC	Sets link condition
Config	General FCTx FCRx TLP AckNak Link Transactions	Performs configuration of generator
Wait	TLP DLLP <# of ns> Error	Performs wait for condition specified

	Link BOB Payload User	
Include	<Include file path>	Includes generator file
Template	TLP DLLP OrderedSet Raw <TemplateName>	Creates template for packet that can be used in Packet command

## 4 Packet command

Initiates transmission of specified packet on the bus.

### 4.1 Packet = TLP

Initiates transmission of TLP packet on the bus.

Parameters

Parameter	Values	Default Value	Comment
PSN	0-0xFFFF, Incr	0	When "Incr" is specified, the PSN for current TLP is assigned as the PSN of previously sent TLP incremented by 1
TLPTType	Mrd32 Mrdlk32 Mwr32 Mrd64 Mrdlk64 Mwr64 Iord Iowr Cfgrd0 Cfgrd1 Cfgrd1 Cfgr1 Msg Msgd Cpl Cpllk Cpld Cpldlk (XX:XXX)	0	Sets Fmt and Type fields in TLP header. When specified in (XX:XXX) format the Fmt and Type fields are set explicitly.
TC	0:7	0	
TD	0:1	0	
EP	0:1	0	
Snoop	0:1	0	
Ordering	0:1	0	
Length		0	If not specified this field is 1 for all read requests and calculated according to payload for write

			requests
Tag		0	
RequesterID	(XX:XX:X) or direct value	0	(BusNumber:DeviceNumber:FunctionNumber)
ECRC			When not specified, the ECRC is calculated automatically. (This assumes that the TD field has been specified.)
LCRC			When not specified then calculated automatically
Payload	(XXXX,XXXX,...)		Specified as DWORDs in hex format, Big Endian.
RawData@<start>			Inserts raw data symbols at <start> position, see packet=Raw description for possible raw data formats
Count	1:0xFFFF		Repeats this packet. Same functionality as repeat command applied to a single packet.

#### 4.1.1 Type= Mrd32, Mrdlk32, Mwr32

Parameter	Value	Default	Comment
LastDwBe	0-0xF	0	
FirstDwBe	0-0xF	0	
Address		0	

#### 4.1.2 Type= Mrd64, Mrdlk64, Mwr64

Parameter	Value	Default	Comment
LastDwBe	0-0xF	0	
FirstDwBe	0-0xF	0	
AddressLo		0	
AddressHi		0	

#### 4.1.3 Type= Iord, Iowr

Parameter	Value	Default	Comment
LastDwBe	0-0xF	0	
FirstDwBe	0-0xF	0	
Address		0	

#### 4.1.4 Type= Cfgrd0, Cfgwr0, Cfgrd1, Cfgwr1

Parameter	Value	Default	Comment
LastDwBe	0-0xF	0	
FirstDwBe	0-0xF	0	
DeviceID	(XX:XX:X) or direct value	0	
Register		0	

#### 4.1.5 Type= Msg, Msgd



Parameter	Value	Default	Comment
MessageRoute	ToRootComplex ByAddress ByID FromRootComplex Local Gather		
MessageCode	Assert_INTA Assert_INTB Assert_INTC Assert_INTD  Deassert_INTA Deassert_INTB Deassert_INTC Deassert_INTD  PM_Active_State_Nak PM_PME PME_Turn_Off PME_TO_Ack  ERR_COR ERR_NONFATAL ERR_FATAL  Unlock  Set_Slot_Power_Limit  Vendor_Defined_Type0 Vendor_Defined_Type1  Attention_Indicator_On Attention_Indicator_Blink Attention_Indicator_Off Power_Indicator_On Power_Indicator_Blink Power_Indicator_Off Attention_Button_Pressed		
AddressHi			Used only if MessageRoute=ByAddress
AddressLo			Used only if MessageRoute=ByAddress
DeviceID	(XX:XX:X) or direct value		Used only if MessageRoute=ById

#### 4.1.6 Type= Cpl, Cpllk, Cpld, Cpldlk

Parameter	Value	Default	Comment
CompleterId	(XX:XX:X) or direct value		
ComplStatus	SC UR CRS CA	SC	

BCM		0	
ByteCount		0	
LowerAddr		0	

Example:

```
Packet = TLP
{
    PSN = 4
    Type = MWr32
    TC=1
    Snoop=1
    Ordering=1
    Length=32
    RequesterID=(12:3:4)
    Tag=4
    LastDwBe=1
    FirstDwBe=3
    Address=0x1234
    Payload=(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)
}
```

Note that in this example TD and EP fields would be 0.

## 4.2 Packet = DLLP

Initiates transmission of DLLP packet on the bus.

Parameters:

Parameter	Values	Comment
DLLPType	Ack Nak InitFC1_p InitFC1_np InitFC1_cpl InitFC2_p InitFC2_np InitFC2_cpl UpdateFC_p UpdateFC_np UpdateFC_cpl PM_Enter_L1 PM_Enter_L23 PM_Active_State_Request_L1 PM_Request_Ack Vendor	
CRC		When not specified then calculated automatically
RawData@<start>		Inserts raw data symbols at <start> position, see packet=Raw description for possible raw data formats
Count	1:0xFFFF	Repeats this packet. Same functionality as repeat command applied to a single packet.

### 4.2.1 DLLPType = Ack, Nak

Parameter	Values	Default	Comment
AckNak_SeqNum	0-0xFFF	0	

### 4.2.2 DLLPType = InitFC1\_p, InitFC1\_np, InitFC1\_cpl, InitFC2\_p, InitFC2\_np, InitFC2\_cpl, UpdateFC\_p, UpdateFC\_np, UpdateFC\_cpl

Parameter	Values	Default	Comment
VC_ID			
HdrFC			
DataFC			

### 4.2.3 DLLPType = Vendor

Parameter	Values	Default	Comment
Data			Vendor specific data

Example 1:

```
Packet = DLLP
{
  DLLPType = Ack
  AckNak_SeqNum = 0x14
  CRC = 0x3616
}
```

Example 2:

```
Packet = DLLP
{
  DLLPType = InitFC1_P
  VC_ID = 0x0
  HdrFC = 0xA
  DataFC = 0x14
}
```

Here CRC will be calculated automatically

## 4.3 Packet = OrderedSet

Initiates transmission of ordered set on the bus.

Parameters:

Parameter	Values	Default	Comment
SetType	TS1 TS2 FTS Pattern Idle Skip		
RawData@<start>			Inserts raw data symbols at <start> position, see packet=Raw description for possible raw data formats
Count	1:0xFFFF		Repeats this packet. Same functionality as repeat command applied to a single packet.

### 4.3.1 SetType = TS1, TS2

Parameter	Values	Default	Comment
LinkNumber	0-255, PAD	PAD	
LaneNumber	0-31, PAD	PAD	
N_FTS	0-255	0	
TrainingControl	(X,X,X,X)		<b>(PETrainer EML only).</b> Training control bits. The order of the bits is as follows: ("HotReset," "DisableLink," "Loopback," "DisableScrambling")
Identifier	(X,X,X...)		Use the same format as in Packet=RawData with exception of 10-bit codes

In case of 4x setup the keys listed above applies for all lines. When you want to specify line specific parameters, specify lane number as the following:

<key>@<lane\_number> = <value>

Example:

N\_FTS = 255  
- sets N\_FTS equal to 255 for all lanes

N\_FTS@1 = 255  
- sets N\_FTS equal to 255 for lane 1

## 4.4 Packet = Raw

Initiates transmission of raw data on the bus.

Parameters:

Parameter	Values	Default	Comment
RawData	(X,X,X...)		Specifies the array of symbols to send

The elements of data can be specified in the following formats:

1) Symbols:

```
Packet = Raw
{
  RawData = ( K28.5, D21.5, K28.5, D10.2 )
}
```

2) Bytes in hexadecimal format with preceding K/D modifier:

```
Packet = Raw
{
  RawData = ( KBC, DB5, KBC, D4A )
}
```

In addition to generate fully qualified 10 bit symbols you can specify running disparity sign for each symbol:

```
Packet = Raw
{
  RawData = ( K28.5+, D21.5-, K28.5-, D10.2- )
}
```

or specify 10 bit symbols in binary, hex or decimal format:

```
Packet = Raw
{
  RawData = ( 0b0011111010, 0b1100111001, 0b0011111010, 0b1110000110 )
}
```

## 4.5 Packet = <TemplateName>

Initiates transmission of the packet specified by Template command.  
User can override packet fields according to template.

Example:

```
Template = TLP
{
  Name = TestPacket
  Type = MemRd32
  TC = 0
  Tag = 0
  RequesterID = (1:0:0)
  Length = 64
  Address = 0
}
```

```

Packet = TestPacket
{
}

Packet = TestPacket
{
    Address = 64
}

Packet = TestPacket
{
    Address = 128
}

```

This sequence will issue three memory read requests.

## 5 Idle command

Command	Modifiers	Comment
Idle	<# of ns>	Sends idle symbols (D0.0)

## 6 Link command

### 6.1 Link = L0

Transitions link to L0 state.

### 6.2 Link = L1

Transitions link to L1 low power state. Applies only in L0 state.

### 6.3 Link = L0s

Transitions link to L0s low power state. Applies only in L0 state.

### 6.4 Link = Disabled

Force LTSSM to enter "Disabled" state. Applies only in Configuration\_linkwidth\_start or Recovery States.

### 6.5 Link = HotReset

Forces Hot Reset generation. Applies only in Recovery state.

## 6.6 Link = Recovery

Transitions link to Recovery state. Applies only from L0, LOS, or L1 States.

## 6.7 Link = Detect

Transitions link to Detect state. Applies from ANY state.

## 6.8 Link = LTSSMOff

Disables LTSSM.

## 6.9 Link = InitFC

Starts flow control initialization state machine.

# 7 Config command

Configures generator.

## 7.1 Config = General

This command should precede any statement in generator file. There should be only one Config = General command in generator file. All Config = General commands from included files are ignored.

Parameter	Values	Default	Comment
AutoDetect	Yes, No	Yes	Automatically detect link parameters
LinkWidth	1,4	1	Ignored if AutoDetect is set
DirectionRx	Upstream, Downstream	U	
DisableScrambleTx	Yes, No	No	Ignored if AutoDetect is set
DisableDescrambleRx	Yes, No	No	Ignored if AutoDetect is set
ReverseLinesTx	Yes, No	No	Ignored if AutoDetect is set
ReverseLinesRx	Yes, No	No	Ignored if AutoDetect is set
InvertPolarityTx	(X,X,X,X)		Ignored if AutoDetect is set
InvertPolarityRx	(X,X,X,X)		Ignored if AutoDetect is set
BaseSpec10	Yes, No	No	
SkewTx	(X,X,X,X)		
TrainerReset	Yes, No	No	Adding 'TrainerReset = Yes' to a script causes the PETrainer EML to perform the following three resets prior to script execution: a DCM reset on Tx FPGA, a normal reset on Tx FPGA, and a normal reset on Rx FPGA.
UseExtRefClock	Yes, No	Yes	Use external reference clock



## 7.2 Config = FCTx

Allows user to specify number of credits according to which the TLPs are being sent.

Parameter	Values	Default	Comment
CareForFC	Yes,No	Yes	When not set the TLPs are being sent without the regard of how many credits are available

## 7.3 Config = FCRx

Configures automatic UpdateFC DLLP generation.

Parameter	Values	Default	Comment
Timer	In ns (rounded to nearest 8), Off	4200	Periodic timer that controls the sending of UpdateFC DLLPs
PH	0-255	1	Posted Request Headers
NPH	0-255	1	Non-Posted Request Headers
CplH	0-255	1	Completion Headers
PD	0-4095	1024	Posted Request Data Payload
NPD	0-4095	1	Non-Posted Request Data Payload
CplD	0-4095	1024	Completion Data Payload

## 7.4 Config = TLP

Data integrity control.

Parameter	Values	Default	Comment
AutoSeqNumber	Yes, No	Yes	If set to 0, overrides automatic generation of the TLP sequence number, and uses user-defined value of the field in the Packet=TLP commands.
ReplayTimer	In ns (rounded to nearest 8), Off	4200	Timeout in TLP transmitter path that counts time since last Ack or Nak DLLP is received. If set, automatically retransmit TLPs that were NAKed or on replay timer expiration.
AutoRetrain	Yes, No	Yes	If set, enable automatic retraining of the link in case the number of retransmitted TLP is 4. Valid only when AutoRetransmission is set.
TagGeneration	Manual, Default Extended Phantom1 Phantom2 Phantom3	Manual	(TagGeneration parameter is for PETrainer EML only). Tag generation policy for posted TLPs: "Manual" - tags are taken from the script "Default" - use lower 5-bit of Tag field. Zero out higher 3 bits. "Extended" - use 8-bit of Tag field. "Phantom1" - use 1 most significant bit of Function field, and 8-bit of Tag "Phantom2" - use 2 most significant bits of Function field, and 8-bit of Tag "Phantom3" - use 3 bits of Function field, and 8-bit of Tag

## 7.5 Config = AckNak

Parameter	Values	Default	Comment
AckNak	Auto Ack Nak Disable	Auto	“Auto” - automatic ACK/NAK (default) “Ack” - always ACK “Nak” - always NAK “Disable” - disable automatic ACK/NAK DLLP generation
Delay	In ns (rounded to nearest 8)	0	Timer that controls how much delay is added to Ack/Nak DLLP response after TLP reception. Valid if AckNak is Auto, Ack or Nak

## 7.6 Config = Transactions

Parameter	Values	Default	Comment
AutoCfgCompletion	Yes, No	No	If set, automatically handles Configuration Read and Write TLP transactions. For Configuration Read transaction, Completion TLP contains the data read from the internal Configuration Space according to specified register address. For Configuration Write transaction, internal Configuration Space is updated at the address with the data taken from Configuration Write TLP, and Configuration Write Completion is returned.
CfgSpaceFile			Specifies file path to configuration space

## 7.7 Config = Link

Parameter	Values	Default	Comment
FTSCount	0-255	255	Number of FTS ordered sets required (as sent in TS)
ExtendedSynch	Yes, No	Yes	This bit when set forces the transmission of 4096 FTS ordered sets
SkipTimer	In ns (rounded to nearest 8), Off	4720	Periodic timer that controls sending of SKIP ordered sets at specific intervals. Timer's value is measured in lus units

## 8 Wait command

Yields script execution until condition specified is true or timeout expired.

Parameter	Values	Default	Comment
Timeout			Timeout in nanoseconds, 0 means infinite timeout
Display			Message that will be displayed during the waiting in status bar

## 8.1 Wait = TLP

Waits for TLP that matches defined condition. Only TLP Header fields can be specified.

All parameters from packet = TLP instruction are valid except PSN, ECRC, LCRC, Payload.

TLP Header fields can be masked using the following format:

“0x0XAXX” – for hexadecimal values

“0b0001XX” – for binary values

Example:

```
Wait = TLP
{
    TLPType = CfgWr
    Register = "0x1XXX"
    Timeout = 0
}
```

This command will wait infinitely for the configuration write request for all registers from 0x1000 to 0x1FFF.

A count parameter can be applied to this command, which will cause it to wait for that number of packets.

## 8.2 Wait = DLLP

Waits for DLLP that matches defined condition.

All parameters from packet = DLLP instruction are valid except CRC field.

DLLP fields can be masked using the following format:

“0x0XAXX” – for hexadecimal values

“0b0001XX” – for binary values

Example:

```
Wait = DLLP
{
    DLLPType = Ack
    Timeout = 256
}
```

This command will wait for Ack DLLP. The execution will continue if Ack DLLP is received or 256ns timeout expired.

A count parameter can be applied to this command, which will cause it to wait for that number of packets.

## 8.3 Wait = Error

Parameter	Values	Default	Comment
Errors	DLLPCRC, TLPLCRC Delimiter Disparity Symbol, IdleData, SkipLate OrdSetFormat, EndBadPacket		The list of errors to wait for. If not specified this will wait for any error

Example:

```
Wait = Error
{
  Errors = (Delimiter, Disparity, Symbol)
  Timeout = 1024
}
```

A count parameter can be applied to this command, which will cause it to wait for that number of errors.

## 8.4 Wait = Link

Parameter	Values	Default	Comment
Conditions	SKIP, IDLE, TS1, TS2, FTS, PATN, DLLP, TLP, COMMA		The list of conditions to wait for.
TrainingControl	(X,X,X,X)		<b>(PETrainer EML only).</b> Training control bits. The order of the bits is as follows: ("HotReset," "DisableLink," "Loopback," "DisableScrambling")

Example:

```
Wait = Link
{
  Conditions = (COMMA)
  Timeout = 1024
}
```

A count parameter can be applied to this command, which will cause it to wait for that number of packets.

## 8.5 Wait = BOB

Waits for Breakout Board Data match.

Parameter	Values	Default	Comment
Data			Mask and Match for four bits of Breakout Board Data

Example:

```
Wait = BOB
{
  Data = "0b0XXX"
  Timeout = 1024
}
```

A count parameter can be applied to this command, which will cause it to wait for that number of matches to the BOB Data.

## 8.6 Wait = Payload

Waits for TLP payload match.

Parameter	Values	Default	Comment
Data			Mask and Match for up to four dwords of TLP payload (ML - any offset from the

			beginning of payload, EML - zero offset from the beginning of payload)
Data@<offset>			Mask and Match for up to four dwords of TLP payload starting from <offset> offset from the beginning of payload (PETrainer EML only)

Example:

```
Wait = Payload
{
  Data = ( 0xABCDXXXX, 0XXXXXXXXXX, 0XXXXXXXXXX, 0XXXX1234 )
  Timeout = 1024
}
```

```
Wait = Payload
{
  Data@2 = ( 0xABCDXXXX )
  Data@8 = ( 0XXXX1234, 0x5678XXXX )
  Timeout = 1024
}
```

## 8.7 Wait = User

Waits for user input.

## 8.8 Additional “Wait” modifiers

1. Wait = <number>

Unconditionally yields script execution for specified number of nanoseconds.

Example:

```
Wait = 500
```

2. Wait = <Text>

Equivalent to

```
Wait = User
{
  Display = <Text>
}
```

Example:

```
Wait = “Press the button to continue script execution”
```

A count parameter can be applied to this command, which will cause it to wait for that number clicks on the user input button.

## 9 Template command

Creates template for packet that can be used in Packet command. The fields specified in the Template command may be overridden in Packet command.

Example:

```
Template = TLP
{
```

```

    Name = "TestPacket"
    Type = MemRd32
    TC = 0
    Tag = 0
    RequesterID = (1:0:0)
    Length = 64
    Address = 0
}

Packet = TestPacket
{
}

Packet = TestPacket
{
    Address = 64
}

Packet = TestPacket
{
    Address = 128
}

```

This sequence will issue three memory read requests.

## 10 Loop command

This command causes the PETrainer BusEngine to re-execute a block of commands a predefined number of times. Note: Loops require up to 1us to branch to the beginning of the loop. During this time the script execution is paused. Internally generated packets such as SKIP ordered sets, ACK/NAK DLLP's, and flow control updates will still occur as programmed. Loops can be nested up to 4 deep.

Parameter	Values	Default	Comment
Count	0:0xFFFF Infinite		Setting to 0 will also cause an infinite loop

### 10.1 Loop=Begin

This command marks the beginning of the loop.

### 10.2 Loop=End

This command marks the end of the loop.

## 11 Repeat command

This command causes one or more commands to be repeated. This is not implemented as a branch instruction in the BusEngine, but is rather an unfolding or replication of commands during script compilation in the software. This allows back-to-back execution of these commands with as little as 0 symbol times of IDLE traffic between them. This command will increase the size of the script object that is downloaded to the UPAS and will increase download time accordingly.

Parameter	Values	Default	Comment
Count	1:0xFFFF		Values of Infinite and 0 are not supported
Counter			

## 11.1 The Counter Parameter

Any string literal can be used for the "Counter" parameter.

The value of "Counter" parameter can be used within the scope of the "Repeat" statement (i.e. between Repeat=Begin and Repeat=End) in arithmetic expressions for any parameter.

The value of "Counter" parameter changes from 0 to the value of the "Count" parameter minus one.

The arithmetic expressions must be included in round brackets.

The operators are: +, -, \*, /, <<, >>, &, |, ~.

### Example 1:

```
Repeat=Begin { Count=4 Counter=ppp }

    Packet=TLP {
        TLPType = MRd64
        Tag = ( ppp + 0x10 )
        AddressHi = ( 0x400000 + 4 / ( 5 - ppp ) )
    }

Repeat=End
```

Within the scope of this repeat "ppp" can be used in arithmetic expressions for any packet field. The value of "ppp" changes from 0 to 3 in the example.

### Example 2: (nested counters)

```
Repeat=Begin { Count=3 Counter=qqq }

    Packet=DLLP { DLLPType=Ack
        AckNak_SeqNum = ( qqq + 1 ) }

    Packet=DLLP { DLLPType=Ack
        AckNak_SeqNum = ( 0xf & ~qqq ) }

    Repeat=Begin { Count=4 Counter=www }

        Packet=TLP { TLPType=MRd64
            AddressHi = ( 0x400000 + www * 4 + qqq ) }

        Repeat=End

Repeat=End
```

## 11.2 Repeat=Begin

This command marks the beginning of the code being repeated.

## 11.3 Repeat=End

This command marks the end of the code being repeated.

## 12 Branch command

Enables/disables interrupt for the condition specified.

### 12.1 Branch = <condition>

Enables the interrupt for the condition specified.

The conditions are the same as in "Wait" command except "User".

The parameter list is the same as for "Wait" command except "Timeout", "Display" and "Count" parameters.

Parameter	Values	Default	Comment
Name			Name of the branch, need to be specified if this branch is to be disabled later
Proc			Name of the procedure to execute when branch conditions are met

#### Example:

...

```
Proc = Begin
{
  Name ="Procedure1"
}
```

...

```
Proc = End
      ; the following statement specifies that if Delimiter, Disparity or Symbol error occurs
      ; then the code declared in "Procedure1" should be executed
```

```
Branch = Error
{
  Name = "SomeErrorBranch"
  Proc = "Procedure1"
  Errors = (Delimiter, Disparity, Symbol)
}
```

...

```
; disable the branch "SomeErrorBranch" that is specified above
```

```
Branch = Disable
{
  Name = "SomeErrorBranch"
}
```

...

### 12.2 Branch = Disable

Disables the interrupt that was previously enabled.

Parameter	Values	Default	Comment
Name			Name of the branch



## 13 Proc command

Declares the procedure to be executed for "Branch" command. Procedure declaration must proceed its usage in the "Branch" statement.

### 13.1 Proc = Begin

Declares the start point of the procedure.

Parameter	Values	Default	Comment
Name			Name of the procedure

### 13.2 Proc = End

Declares the end point of the procedure.

## 14 Include command

Includes generator file inline. All commands in the included file will be executed with the exception of Config=General command.

# How to Contact LeCroy

Type of Service	Contact
Call for technical support...	US and Canada: 1 (800) 909-2282 Worldwide: 1 408-653-1260
Fax your questions...	Worldwide: 1 (408) 727-6622
Write a letter ...	LeCroy Protocol Solutions Group Customer Support 3385 Scott Blvd. Santa Clara, CA 95054
Send e-mail...	<a href="mailto:support@CATC.com">support@CATC.com</a>
Visit LeCroy's web site...	<a href="http://www.CATC.com/">http://www.CATC.com/</a>