

# iSCSISim

Test Tool for iSCSI Targets

**David J. Cuddihy**  
Principal Engineer  
ATTO Technology, Inc.

# Another iSCSI Test Platform?

- ❑ Why do we need yet another test tool?
  - ❑ Commercial testing platforms don't match our testing workflow
  - ❑ Not much new development in commercial tools
  - ❑ Last year's iSCSI plugfest
    - ❑ We discussed possibility of conformance suite

- **iSCSISim Benefits/Uses**
- Getting Started with iSCSISim
- Configure iSCSISim for testing
- Example
- Coming Attractions

On later slides the non-current agenda items are greyed out.

# What's Better about iSCSISim

- ❑ Full scripting Capability
  - ❑ No user intervention
- ❑ Automatic iterations
- ❑ No data restrictions
- ❑ Obvious indication of test pass/fail

- ❑ Transport-Level Commands
  - ❑ With/Without Digests
  - ❑ Auto Generation of Sequence Numbers
- ❑ SCSI Commands
  - ❑ With/Without Data
  - ❑ Auto Generate ITT, CmdSn, TTT, Digests
- ❑ Check Responses
  - ❑ Auto Generation of Sequence Numbers
  - ❑ Fields/Data may be ignored

- ❑ Testing of Improperly Formatted Commands
  - ❑ Login problems
  - ❑ Text Messages
  - ❑ Bad BHS
  - ❑ Bad Checksum
  - ❑ Wrong Sequence Numbers
  - ❑ etc.

# What's the Use of ISCSISim?

- ❑ Developer (unit-level) Testing, allowing:
  - ❑ Fine-grained control of PDU content
  - ❑ Automatic verification of unit tests
  - ❑ Ability to Send poorly formatted PDU's
  - ❑ Ability to Recreate bug from a trace
- ❑ Regression Testing
- ❑ Conformance Testing
  - ❑ Automatic validation of results

# Opportunities to improve iSCSI

- ❑ Performance testing
  - ❑ Not the ‘right tool’ for the ‘right job’
- ❑ Multi-initiator tests
- ❑ Multi-connection tests



- iSCSI Benefits/Uses
- **Getting Started with iSCSI**
- Configure iSCSI for testing
- Example
- Coming Attractions

# Example : Login, Write/Verify

```
<ISCSITS iterations = "1">
  <FILENAME>
    ITDLogin.xml
  </FILENAME>
  <FILENAME>
    ITDClearUnitAtt.xml
  </FILENAME>
  <FILENAME>
    ITDWriteRead64K.xml
  </FILENAME>
</ISCSITS>
```

- ❑ Simple Test Case
  - ❑ Login
    - ❑ full feature login
  - ❑ Clear Unit Attentions
  - ❑ Write 64K
    - ❑ 64K immediate data
  - ❑ Read Back 64k
    - ❑ compare data

# iSCSISim : Getting Started

- ❑ Instructions on SourceForge
  - ❑ <https://sourceforge.net/projects/iscsisim>
- ❑ Python 2.5
  - ❑ <http://www.python.org>
  - ❑ Additional Packages:
    - ❑ wxPython : <http://wxpython.org>
    - ❑ ElementTree : <http://effbot.org/downloads/#elementtree>
    - ❑ pySerial : <http://pyserial.sourceforge.net>
- ❑ iSCSISim
  - ❑ Source code
  - ❑ Example Test cases

## Config.txt file for tool customization

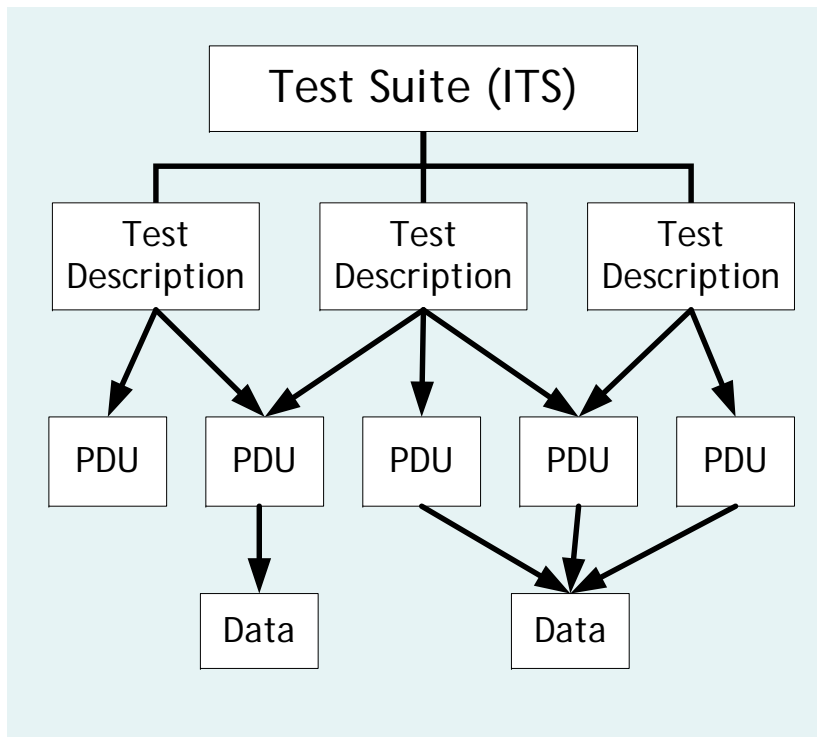
```
[PROJECT]
comms = TCP
targetname = iqn.2008-2.com.po:Sn12345
[TCP]
IpAddress = 10.30.45.10
TcpPort = 3260
```

- ❑ Comms can be Serial or TCP
- ❑ Targetname (optional)
  - ❑ string substitution
  - ❑ fixes up data lengths

- iSCSISim Benefits/Uses
- Getting Started with iSCSISim
- **Configure iSCSISim for testing**
- Example
- Coming Attractions

# Test Specification Hierarchy

Tests are specified via XML files. Hierarchical structure provides flexibility and easy reuse of PDU descriptions.



- Test Suite
  - Iterable
- Test Description
  - List of PDU's
  - Execution criteria
- PDU File
  - Single Protocol Data Unit
- Data File
  - Binary data

# PDU Description File - 64K write

```
<ISCSIPDU>
  <BHS>
    <DWORD1>01A00000</DWORD1>
    <AHSLEN>00</AHSLEN>
    <DATASEGLLEN>010000</DATASEGLLEN>
    <LUN>0000000000000000</LUN>
    <ITT>00000003</ITT>
    <DW20>00010000</DW20>
    <DW24>$CMD_SN</DW24>
    <DW28>$EXP_STAT_SN</DW28>
    <DW32>2A000000</DW32>
    <DW36>02000000</DW36>
    <DW40>80000000</DW40>
    <DW44>00000000</DW44>
  </BHS>
  <!-- AHS is not represented -->
  <HEADERDIGEST />
  <DATA len="10000">
    <FILE>Binaries\Data64K.bin</FILE>
  </DATA>
  <DATADIGEST />
</ISCSIPDU>
```

```
<!-- basic header segment start -->
<!-- operation code -->
<!-- additional header seg. length -->
<!-- data segment length -->
<!-- logical unit number -->
<!-- initiator task tag -->
<!-- expected data trans. length -->
<!-- command sequence number -->
<!-- expected status sequence num -->
<!-- command descriptor block start -->
<!-- LBA = 512 -->
<!-- write 0x80 blocks (64K) -->
<!-- command descriptor block end -->
<!-- basic header segment end -->

<!-- data segment start -->
<!-- immediate data -->
<!-- data segment end -->
```

# Definition of PDU Keywords

Name	TX	RX	Notes
\$CMD_SN	X	X	Automatically generated for both Send & Receive PDUs
\$EXP_STAT_SN	X		Automatically Generated
\$ITT	X	X	Automatically Generated
\$TTT	X		Scanned from R2T
\$DIGEST	X	X	Digest is automatically computed
\$X		X	Don't care – don't check this field



- ❑ Two Digest Elements in PDU Description File
  - ❑ HEADERDIGEST
  - ❑ DATADIGEST
- ❑ Possible Values
  - ❑ If element is null, CRC is not inserted into stream
  - ❑ If element is \$DIGEST, automatically calculate digest
  - ❑ Use a Hex string to create a bad digest for testing
- ❑ Digests are tested during receive PDU validation
  - ❑ If digest was not null on send, will be tested
  - ❑ No provision to ignore digest (future enhancement)

- ❑ Data is specified as a *binary* file
  - ❑ Specify filename, length
- ❑ Pad is automatically added
  - ❑ Do not adjust XML PDU description (length/expected data transfer length) for pad
- ❑ Login substitution
  - ❑ \$TGTNAME\$ will be replaced with string from config file
  - ❑ ExpectedDataTransferLength will be automatically updated

# Test Description File : ITD

```
<ISCSITD>
  <CmdSeqList>
    <CmdSeq Id = "1">
      <PDU i="1">PDUTReqBrg.xml</PDU>
      <PDU i="2">PDUTRspBrg.xml</PDU>
    </CmdSeq>
    <CmdSeq Id = "2">
      <PDU i="1">PDUScsiWrite64K_32KI.xml</PDU>
      <PDU i="2">PDUR2T32K_64K.xml</PDU>
      <PDU i="3">PDUDataOutOOO_1.xml</PDU>
      <PDU i="4">PDUDataOutOOO_2.xml</PDU>
      <PDU i="5">PDUDataOutOOO_3.xml</PDU>
      <PDU i="6">PDUScsiStatusGood.xml</PDU>
    </CmdSeq>
  </CmdSeqList>
  <RunList>
    <R>(1,1)</R>
    <R Trig="(1,2)">(2,1)</R>
    <R Trig="(2,2)">(2,3)</R>
    <R>(2,4)</R>
    <R>(2,5)</R>
    <R Trig="(2,6)">/>
  </RunList>
</ISCSITD>
```

Cmd Sequences : list of PDUs with the same ITT

Cmd Sequence includes test and expected response

Run List : List of tuples referring to order of test

Optional Trigger indicates expected response to wait for prior to sending next request

ITD provides facility for testing multiple outstanding commands

```
<RunList>  
  <R>(1,1)</R>  
  <R Trig="(1,2)">(2,1)</R>  
  <R >(3,1)</R>  
  <R Trig="(2,2)" />  
  <R Trig="(3,2)" />  
</RunList>
```

- ❑ Runlist: specify commands without trigger
- ❑ Order of responses must be predictable – use ordered queuing if necessary

# Test Suite File : ITS

A Test Suite is a list of test description files to be run in order. Number of iterations is specified in the test suite file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Test Suite - do a login, NOP-OUT, logout -->
<!-- Execute the login test suite -->
<ISCSITS iterations="1">
  <FILENAME>ITDLogin.xml</FILENAME>
  <FILENAME>ITDNop.xml</FILENAME>
  <FILENAME>ITDLogout.xml</FILENAME>
</ISCSITS>
```

- ❑ Extremely detailed Log
  - ❑ What was sent
  - ❑ What was received
  - ❑ Description of errors (if any)
- ❑ New Log file created for each run
- ❑ Placed in Logfiles directory

- iSCSISim Benefits/Uses
- Getting Started with iSCSISim
- Configure iSCSISim for testing
- **Example**
- Coming Attractions

With Digests enabled, execute the following

- ❑ Login
- ❑ Send write command (no immediate data)
  - ❑ Wait for R2T
  - ❑ Automatically Generate TTT for response
- ❑ Send 2 simultaneous read commands
  - ❑ Ignore data from first read
  - ❑ Check data on second read



# Example Test Suite File

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Test Suite - do a login, Write,
      2 Reads, logout
-->
<!-- Execute the login test suite -->
<ISCSITS iterations="1">
  <FILENAME>ITDLogin.xml</FILENAME>
  <FILENAME>ITDWrite64KNoImmed.xml</FILENAME>
  <FILENAME>ITD2Reads64K.xml</FILENAME>
  <FILENAME>ITDLogout.xml</FILENAME>
</ISCSITS>
```

# ITD For Write

```
<ISCSITD>
  <CmdSeqList>
    <CmdSeq Id = "1">
      <PDU i="1">PDUWrite64KNoImmed.xml</PDU>
      <PDU i="2">PDUR2T64K.xml</PDU>
      <PDU i="3">PDUDataOut64K.xml</PDU>
      <PDU i="4">PDU SCSI Status Good.xml</PDU>
    </CmdSeq>
  </CmdSeqList>
  <RunList>
    <R>(1,1)</R>
    <R Trig="(1,2)">(1,3)</R>
    <R Trig="(1,4)" />
  </RunList>
</ISCSITD>
```

# 64K Write PDU File

```
<ISCSIPDU>
  <BHS>
    <DWORD1>01A00000</DWORD1>
    <AHSLEN>00</AHSLEN>
    <DATASEGLLEN>00</DATASEGLLEN>
    <LUN>0000000000000000</LUN>
    <ITT>00000003</ITT>
    <DW20>00010000</DW20>
    <DW24>$CMD_SN</DW24>
    <DW28>$EXP_STAT_SN</DW28>
    <DW32>2A000000</DW32>
    <DW36>02000000</DW36>
    <DW40>80000000</DW40>
    <DW44>00000000</DW44>
  </BHS>
  <!-- AHS is not represented -->
  <HEADERDIGEST>$DIGEST</HEADERDIGEST>
  <DATA />
  <DATADIGEST />
</ISCSIPDU>
```

*Write(10) CDB*

# Data Out 64K PDU File

```
<ISCSIPDU>
  <BHS>
    <DWORD1>05800000</DWORD1>
    <AHSLEN>00</AHSLEN>
    <DATASEGLLEN>010000</DATASEGLLEN>
    <LUN>0000000000000000</LUN>
    <ITT>00000003</ITT>
    <DW20>$TTT</DW20>
    <DW24>00000000</DW24>
    <DW28>$EXP_STAT_SN</DW28>
    <DW32>00000000</DW32>
    <DW36>00000000</DW36>
    <DW40>00010000</DW40>
    <DW44>00000000</DW44>
  </BHS>
  <!-- AHS is not represented -->
  <HEADERDIGEST>$DIGEST</HEADERDIGEST>
  <DATA len="10000">
    <FILE>Binaries\DataBig.bin</FILE>
  </DATA>
  <DATADIGEST>$DIGEST</DATADIGEST>
</ISCSIPDU>
```

# ITD For Reads

```
<ISCSITD>
  <CmdSeqList>
    <CmdSeq Id = "1">
      <PDU i="1">PDURead64K.xml</PDU>
      <PDU i="2">PDUDataIn64K.xml</PDU>
    </CmdSeq>
    <CmdSeq Id = "2">
      <PDU i="1">PDURead64K_B.xml</PDU>
      <PDU i="2">PDUDataIn64K_B.xml</PDU>
    </CmdSeq>
  </CmdSeqList>
  <RunList>
    <R>(1,1)</R>
    <R>(2,1)</R>
    <R>Trig="(2,2)"</R>
  </RunList>
</ISCSITD>
```

*Send Both Reads Without Waiting*

# 64K Read PDU File

```
<ISCSIPDU>
  <BHS>
    <DWORD1>41400000</DWORD1>
    <AHSLEN>00</AHSLEN>
    <DATASEGLLEN>000000</DATASEGLLEN>
    <LUN>0000000000000000</LUN>
    <ITT>$ITT</ITT>
    <DW20>010000</DW20>
    <DW24>$CMD_SN</DW24>
    <DW28>$EXP_STAT_SN</DW28>
    <DW32>28000000</DW32>
    <DW36>02000000</DW36>
    <DW40>80000000</DW40>
    <DW44>00000000</DW44>
  </BHS>
  <!-- AHS is not represented -->
  <HEADERDIGEST>$DIGEST</HEADERDIGEST>
  <DATA />
  <DATADIGEST />
</ISCSIPDU>
```

*READ (10) CDB*

# 64K Data in (B) file

```
<ISCSIPDU>
  <BHS>
    <DWORD1>25810000</DWORD1>
    <AHSLEN>00</AHSLEN>
    <DATASEGLLEN>010000</DATASEGLLEN>
    <LUN>0</LUN>
    <ITT>$ITT</ITT>
    <DW20>$X</DW20>
    <DW24>$X</DW24>
    <DW28>$X</DW28>
    <DW32>$X</DW32>
    <DW36>$X</DW36>
    <DW40>0</DW40>
    <DW44>0</DW44>
  </BHS>
  <!-- AHS is not represented -->
  <HEADERDIGEST>$DIGEST</HEADERDIGEST>
  <DATA len="10000" compare = "True">
    <FILE>Binaries\DataBig.bin</FILE>
  </DATA>
  <DATADIGEST>$DIGEST</DATADIGEST>
</ISCSIPDU>
```

*"Don't Cares"*

- iSCSISim Benefits/Uses
- Getting Started with iSCSISim
- Configure iSCSISim for testing
- Example
- **Coming Attractions**



There are large and small modifications on the iSCSISim roadmap

- ❑ Large Efforts
  - ❑ Multi Connection Support
  - ❑ PDU Construction GUI
  - ❑ Auto Discovery/Auto Login
- ❑ Small Modifications
  - ❑ Graceful Handling of Response Timeout
  - ❑ Read Performance Improvements
  - ❑ Trigger from Timeout
  - ❑ IPV6

# You Can Help!

- ❑ Please consider joining the project team.
  - ❑ Especially in need of GUI help
- ❑ Contact :
  - ❑ SourceForge e-mail -  
davidcuddihy *\_at\_* users.sourceforge.net
  - ❑ David Cuddihy – ATTO Technology, Inc.