



The bridge to possible

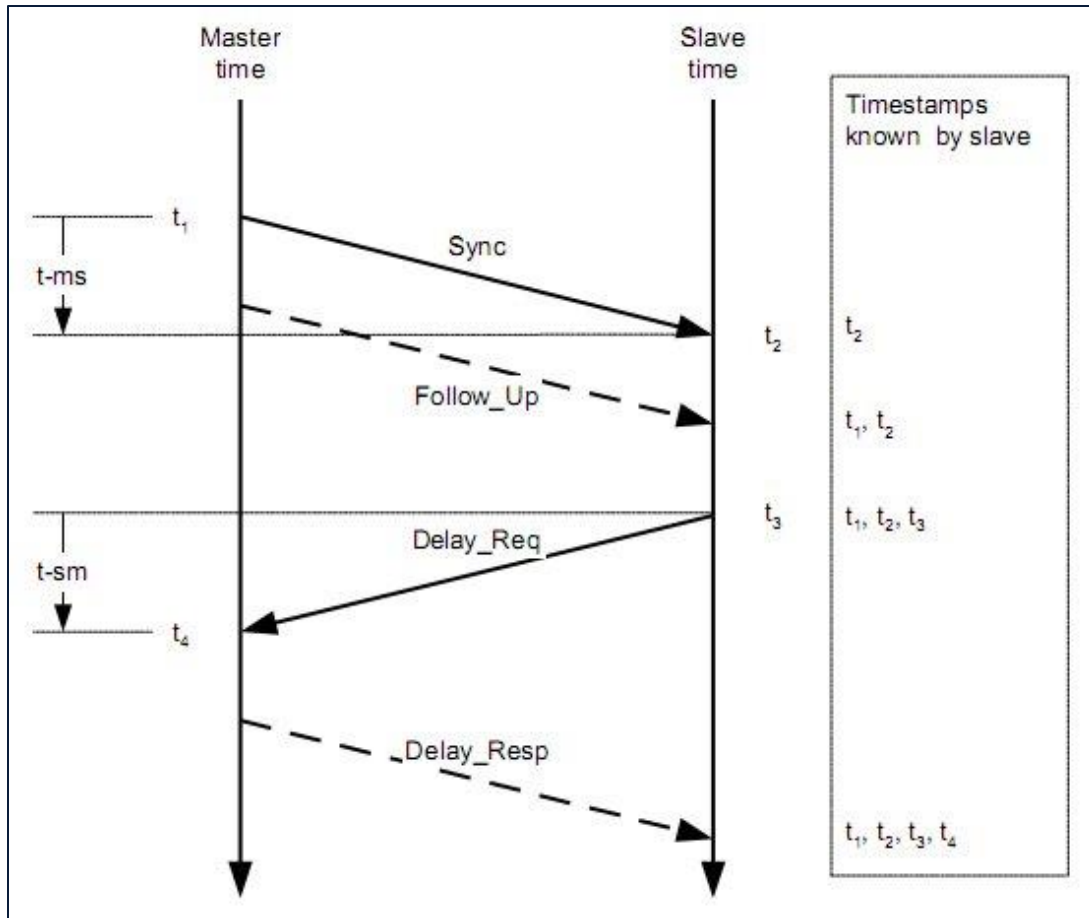
Introduce to PTP Bad timestamp Analyzer(PBA)

An idea from SR

Stone Li(guangli2@cisco.com)



IEEE 1588 PTP Clock Synchronization



- ◆ Offset from Master
$$\text{Offset} = [(t_2 - t_1) - (t_4 - t_3)] / 2$$
- ◆ IF $| \text{Offset} | > 1500\text{ns}$, It is a **bad timestamp**
- ◆ When a bad timestamp is occurring, **PTP Status should be changed** in show log
- ◆ Bad timestamp : calculated from PTP debug log
- ◆ PTP Status change : show log

Enable PTP Debug

Enable PTP Debug

- Enable logging

starts the logging into ptp_trace_file.log


```
debug ptp platform servo log 3
```

- Clear the logfile

```
>/harddisk\ptp_trace_file.log
```

```
RP/0/RP1/CPU0:NCS-5508-A#debug ptp platform servo log 3
wed Jun 16 07:02:07.456 UTC
RP/0/RP1/CPU0:NCS-5508-A#run
[xr-vm_node0_RP1_CPU0:/harddisk:]$>ptp_trace_file.log
```

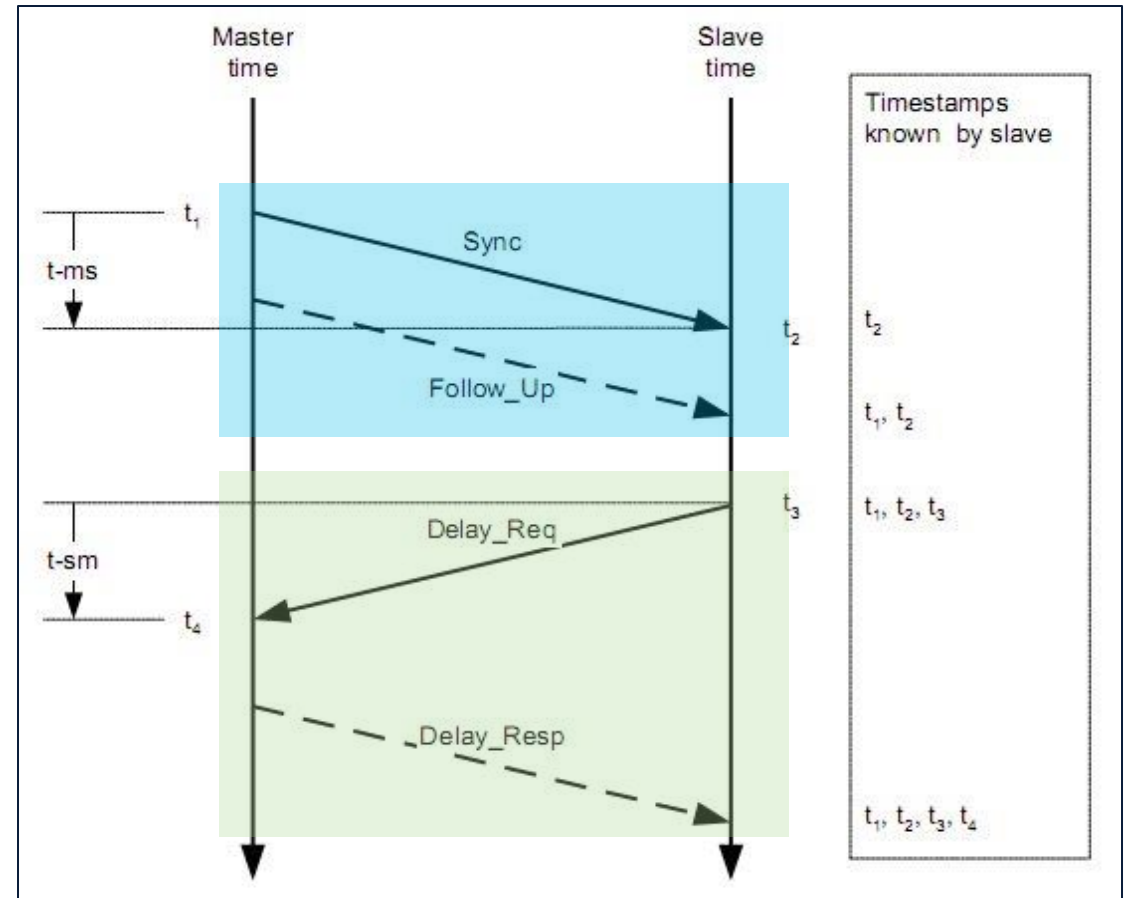
ptp_trace_file.log



```
Raw Input: Tx 00005EAB1214.03810690, Rx 00005EAB1214.0383250F
Raw Input: stream:0
Ts diff (with Corr_sec): Tx 5EAB1214.03810690, Rx 5EAB1214.0383250F, _CorrSec_00000000, _stream:0
Raw Input: Tx 00005EAB1214.046E7238, Rx 00005EAB1214.047090B4
Raw Input: stream:1
Ts diff (with Corr_sec): Tx 5EAB1214.046E7238, Rx 5EAB1214.047090B4, _CorrSec_00000000, _stream:1
[12292689.370806140, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834896
Raw Input: Tx 00005EAB1214.0731BB9C, Rx 00005EAB1214.0733DA16
Raw Input: stream:0
Ts diff (with Corr_sec): Tx 5EAB1214.0731BB9C, Rx 5EAB1214.0733DA16, _CorrSec_00000000, _stream:0
Raw Input: Tx 00005EAB1214.0840E3D8, Rx 00005EAB1214.08430254
Raw Input: stream:1
Ts diff (with Corr_sec): Tx 5EAB1214.0840E3D8, Rx 5EAB1214.08430254, _CorrSec_00000000, _stream:1
[12292689.370842970, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834899
Raw Input: Tx 00005EAB1214.0AC72F14, Rx 00005EAB1214.0AC94D96
Raw Input: stream:0
Ts diff (with Corr_sec): Tx 5EAB1214.0AC72F14, Rx 5EAB1214.0AC94D96, _CorrSec_00000000, _stream:0
Raw Input: Tx 00005EAB1214.0BD438FC, Rx 00005EAB1214.0BD65778
Raw Input: stream:1
Ts diff (with Corr_sec): Tx 5EAB1214.0BD438FC, Rx 5EAB1214.0BD65778, _CorrSec_00000000, _stream:1
[12292689.467516672, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834895
Raw Input: Tx 00005EAB1214.0E972AFC, Rx 00005EAB1214.0E99497F
Raw Input: stream:0
Ts diff (with Corr_sec): Tx 5EAB1214.0E972AFC, Rx 5EAB1214.0E99497F, _CorrSec_00000000, _stream:0
Raw Input: Tx 00005EAB1214.0F84DDC1, Rx 00005EAB1214.0F86FC3C
Raw Input: stream:1
Ts diff (with Corr_sec): Tx 5EAB1214.0F84DDC1, Rx 5EAB1214.0F86FC3C, _CorrSec_00000000, _stream:1
```

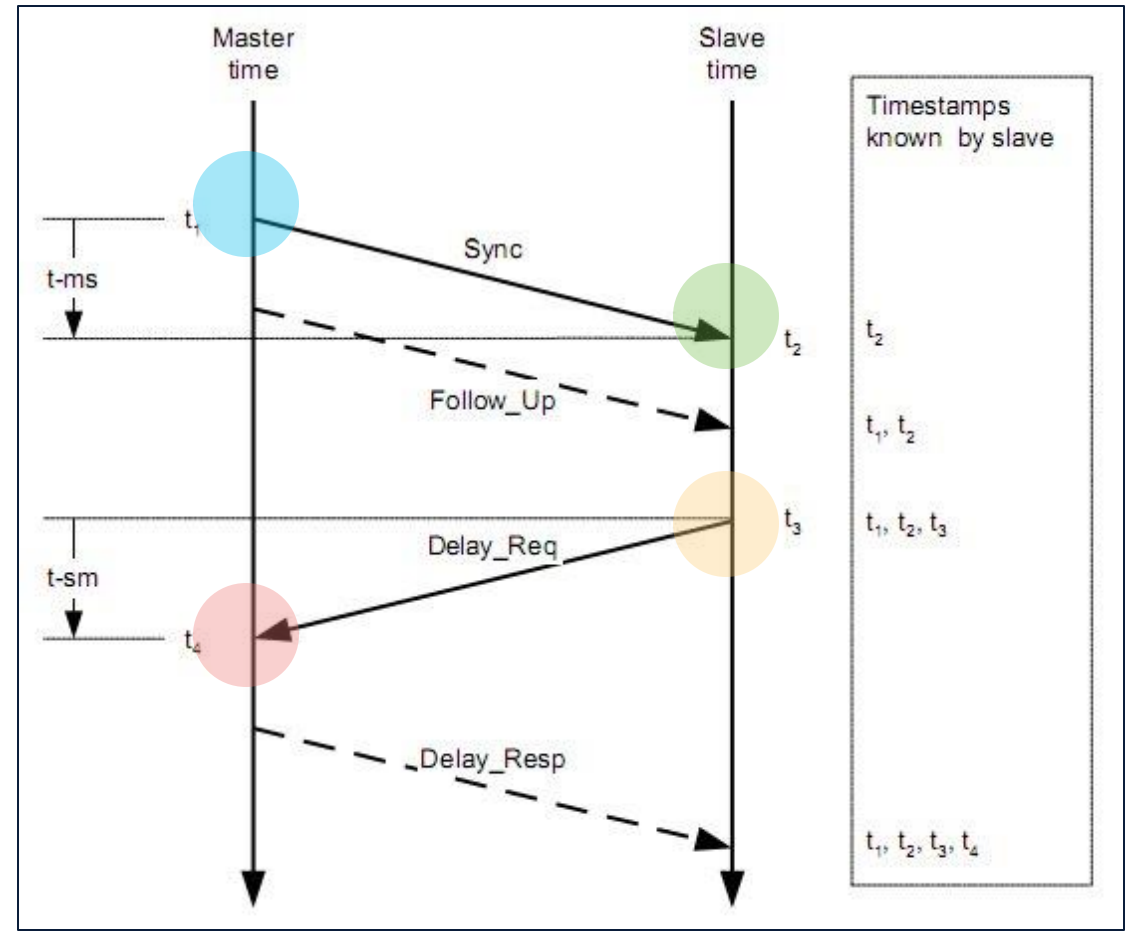
One PTP Clock Synchronization

```
Raw Input: Tx 00005EAB1214.03810690, Rx 00005EAB1214.0383250F
Raw Input: stream:0
Ts diff (with Corr_sec): Tx 5EAB1214.03810690, Rx 5EAB1214.0383250F, CorrSec 00000000, stream:0
Raw Input: Tx 00005EAB1214.046E7238, Rx 00005EAB1214.047090B4
Raw Input: stream:1
Ts diff (with Corr_sec): Tx 5EAB1214.046E7238, Rx 5EAB1214.047090B4, CorrSec 00000000, stream:1
[12292689.370806140, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834896
Raw Input: Tx 00005EAB1214.0731BB9C, Rx 00005EAB1214.0733DA16
Raw Input: stream:0
Ts diff (with Corr_sec): Tx 5EAB1214.0731BB9C, Rx 5EAB1214.0733DA16, CorrSec 00000000, stream:0
Raw Input: Tx 00005EAB1214.0840E3D8, Rx 00005EAB1214.08430254
Raw Input: stream:1
Ts diff (with Corr_sec): Tx 5EAB1214.0840E3D8, Rx 5EAB1214.08430254, CorrSec 00000000, stream:1
[12292689.370842970, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834899
Raw Input: Tx 00005EAB1214.0AC72F14, Rx 00005EAB1214.0AC94D96
Raw Input: stream:0
Ts diff (with Corr_sec): Tx 5EAB1214.0AC72F14, Rx 5EAB1214.0AC94D96, CorrSec 00000000, stream:0
Raw Input: Tx 00005EAB1214.0BD438FC, Rx 00005EAB1214.0BD65778
Raw Input: stream:1
Ts diff (with Corr_sec): Tx 5EAB1214.0BD438FC, Rx 5EAB1214.0BD65778, CorrSec 00000000, stream:1
[12292689.467516672, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834895
Raw Input: Tx 00005EAB1214.0E972AFC, Rx 00005EAB1214.0E99497F
Raw Input: stream:0
Ts diff (with Corr_sec): Tx 5EAB1214.0E972AFC, Rx 5EAB1214.0E99497F, CorrSec 00000000, stream:0
Raw Input: Tx 00005EAB1214.0F84DDC1, Rx 00005EAB1214.0F86FC3C
Raw Input: stream:1
Ts diff (with Corr_sec): Tx 5EAB1214.0F84DDC1, Rx 5EAB1214.0F86FC3C, CorrSec 00000000, stream:1
```



Timestamp(HEX) in PTP Debug file

```
Raw Input: Tx 00005EAB1214.2FBA59CC, Rx 00005EAB1214.2FBC7846
Raw Input: stream:0
Ts diff (with Corr sec): Tx 5EAB1214.2FBA59CC, Rx 5EAB1214.2FBC7846,
Raw Input: Tx 00005EAB1214.30C8B260, Rx 00005EAB1214.30CAD0E0
Raw Input: stream:1
Ts diff (with Corr sec): Tx 5EAB1214.30C8B260, Rx 5EAB1214.30CAD0E0,
[12292690.048054083, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834899
Raw Input: Tx 00005EAB1214.3479282D, Rx 00005EAB1214.347B46B0
Raw Input: stream:1
Ts diff (with Corr sec): Tx 5EAB1214.3479282D, Rx 5EAB1214.347B46B0,
[12292690.144862097, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834899
[12292690.172002379, 0x7ff5dd998ba0:65534] b_4100: 0 0 0
Raw Input: Tx 00005EAB1214.375C9EE0, Rx 00005EAB1214.375EBD62
Raw Input: stream:0
Ts diff (with Corr sec): Tx 5EAB1214.375C9EE0, Rx 5EAB1214.375EBD62,
Raw Input: Tx 00005EAB1214.380CD800, Rx 00005EAB1214.380EF680
Raw Input: stream:1
Ts diff (with Corr sec): Tx 5EAB1214.380CD800, Rx 5EAB1214.380EF680,
[12292690.241486374, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834898
```



Timestamp Explanation

Raw Input: Tx 00005EAB1214.2FBA59CC, Rx 00005EAB1214.2FBC7846

Raw Input: stream:0

Ts diff (with Corr sec): Tx 5EAB1214.2FBA59CC, Rx 5EAB1214.2FBC7846, CorrSec 00000000, stream:0

2020-05-01 01:59:48

800741836

Check PTP Status Log

show logging | in Servo

```
RP/0/RP0/CPU0:May 26 04:03:21.408 KST: ptp_ctrlr[1159]: %PLATFORM-PTP-6-SERVO_EVENTS : PTP Servo state transition from state PHASE_LOCKED to state FREQ_LOCKED
RP/0/RP0/CPU0:May 26 04:03:51.432 KST: ptp_ctrlr[1159]: %PLATFORM-PTP-6-SERVO_EVENTS : PTP Servo state transition from state FREQ_LOCKED to state PHASE_LOCKED
RP/0/RP0/CPU0:May 26 16:56:27.310 KST: ptp_ctrlr[1159]: %PLATFORM-PTP-6-SERVO_EVENTS : PTP Servo state transition from state PHASE_LOCKED to state FREQ_LOCKED
RP/0/RP0/CPU0:May 26 16:56:57.816 KST: ptp_ctrlr[1159]: %PLATFORM-PTP-6-SERVO_EVENTS : PTP Servo state transition from state FREQ_LOCKED to state PHASE_LOCKED
```

PTP Debug File

```
Raw Input: Tx 00005EAB1214.03810690, Rx 00005EAB1214.0383250F
Raw Input: stream:0
Ts diff (with Corr sec): Tx 5EAB1214.03810690, Rx 5EAB1214.0383250F, CorrSec_00000000, stream:0
Raw Input: Tx 00005EAB1214.046E7238, Rx 00005EAB1214.047090B4
Raw Input: stream:1
Ts diff (with Corr sec): Tx 5EAB1214.046E7238, Rx 5EAB1214.047090B4, CorrSec_00000000, stream:1
[12292689.370806140, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834896
Raw Input: Tx 00005EAB1214.0731BB9C, Rx 00005EAB1214.0733DA16
Raw Input: stream:0
Ts diff (with Corr sec): Tx 5EAB1214.0731BB9C, Rx 5EAB1214.0733DA16, CorrSec_00000000, stream:0
Raw Input: Tx 00005EAB1214.0840E3D8, Rx 00005EAB1214.08430254
Raw Input: stream:1
Ts diff (with Corr sec): Tx 5EAB1214.0840E3D8, Rx 5EAB1214.08430254, CorrSec_00000000, stream:1
[12292689.370842970, 0x7ff5dd998ba0:0] No. 2 a_90060 y = 0_821834899
```

2 approaches to calculate a bad timestamp

- Method 1

From show log to PTP Debug File(which is widely used)

- Method 2

From PTP Debug File to show log

Method 1 - Calculate Steps

```
RP/0/RP0/CPU0:May 26 16:56:27.310 KST: ptp_ctrlr[1159]: $PLATFORM-PTP-6-SERVO_EVENTS : PTP Servo state transition from state PHASE_LOCKED to state FREQ_LOCKED
RP/0/RP0/CPU0:May 26 16:56:57.816 KST: ptp_ctrlr[1159]: $PLATFORM-PTP-6-SERVO_EVENTS : PTP Servo state transition from state FREQ_LOCKED to state PHASE_LOCKED
```

1

Web Dev ▾ Conversion ▾ Encoders / Decoders ▾ Formatters ▾ Internet ▾

Enter a Date & Time

Year	Month	Day	Hour (24 hour)	Minutes	Seconds
2021	05	26	15	56	27

Convert →

Unix Timestamp	1622015787
GMT	Wed May 26 2021 07:56:27 GMT+0000
Your Time Zone	Wed May 26 2021 15:56:27 GMT+0800 (China Standard Time)
Relative	23 days ago

2

apidTables

Home ▸ Conversion ▸ Number conversion ▸ Decimal to hexadecimal

Decimal to Hexadecimal converter

From: To:

Enter decimal number

Convert **Reset** **Swap**

Hex number

```
Raw Input: Tx 000060ADFF2B.0146EDAA, Rx 000060ADFF2B.0146EF94
Ts diff (with Corr sec): Tx 60ADFF2B.0146EDAA, Rx 60ADFF2B.0146EF94, CorrSec 00000000, stream:1
Raw Input: Tx 000060ADFF2B.016D3710, Rx 000060ADFF2B.016D3904
Ts diff (with Corr sec): Tx 60ADFF2B.016D3710, Rx 60ADFF2B.016D3904, CorrSec 00000000, stream:0
Raw Input: Tx 000060ADFF2B.04F8F5A0, Rx 000060ADFF2B.04F8F794
Ts diff (with Corr sec): Tx 60ADFF2B.04F8F5A0, Rx 60ADFF2B.04F8F794, CorrSec 00000000, stream:1
Raw Input: Tx 000060ADFF2B.0526E3B0, Rx 000060ADFF2B.0526E59B
Ts diff (with Corr sec): Tx 60ADFF2B.0526E3B0, Rx 60ADFF2B.0526E59B, CorrSec 00000000, stream:0
Raw Input: Tx 000060ADFF2B.08AAB164, Rx 000060ADFF2B.08AAB354
Ts diff (with Corr sec): Tx 60ADFF2B.08AAB164, Rx 60ADFF2B.08AAB354, CorrSec 00000000, stream:1
Raw Input: Tx 000060ADFF2B.08E09050, Rx 000060ADFF2B.08E09237
Ts diff (with Corr sec): Tx 60ADFF2B.08E09050, Rx 60ADFF2B.08E09237, CorrSec 00000000, stream:0
Raw Input: Tx 000060ADFF2B.0C5D0B60, Rx 000060ADFF2B.0C5D0D54
Ts diff (with Corr sec): Tx 60ADFF2B.0C5D0B60, Rx 60ADFF2B.0C5D0D54, CorrSec 00000000, stream:1
Raw Input: Tx 000060ADFF2B.0C9A3CF0, Rx 000060ADFF2B.0C9A3ED7
Ts diff (with Corr sec): Tx 60ADFF2B.0C9A3CF0, Rx 60ADFF2B.0C9A3ED7, CorrSec 00000000, stream:0
```

Hex Calculator

Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:

0146EF94 – 0146EDAA = **1EA**

Decimal value:

21426068 – 21425578 = **490**

= ?

Calculate **Clear**

Evaluate Workload

Need to Hex convert $16 \times 2 = 32$ times and 1 time offset calculation 1 second

	A	B	C	D	E	F	G	H	I
1	Debug_Time	T1	T2	T3	T4	T2-T	T4-T	Offset	BTSYI
11	2021-05-26 15:10:43	310360D2	310361E5	3295D2D4	3295D3F2	275	286	-5.5	
12	2021-05-26 15:10:43	34B70926	34B70A2C	3647F83C	3647F956	262	282	-10	
13	2021-05-26 15:10:43	38685BF6	38685D04	39F9974C	39F99862	270	278	-4	
14	2021-05-26 15:10:44	007F19F2	007F1AFC	0210D544	0210D65E	266	282	-8	
15	2021-05-26 15:10:44	0430D846	0430D950	05C31A88	05C31BA2	266	282	-8	
16	2021-05-26 15:10:44	07E281F6	07E28304	097474F0	09747606	270	278	-4	
17	2021-05-26 15:10:44	0B94935A	0B94946C	0D27E26C	0D27E386	274	282	-4	
18	2021-05-26 15:10:44	0F46A6CA	0F46A7DE	10D90525	10D90636	276	273	1.5	
19	2021-05-26 15:10:44	12F99A62	12F99B7A	148B30D2	148B31DE	280	268	6	
20	2021-05-26 15:10:44	16AA7DFA	16AA7F16	183DA31A	183DA42A	284	272	6	
21	2021-05-26 15:10:44	1A5C87FE	1A5C890C	1BEF5A88	1BEF5BA2	270	282	-6	
22	2021-05-26 15:10:44	1E0E9D3A	1E0E9E48	1FA124A0	1FA125BE	270	286	-8	
23	2021-05-26 15:10:44	21C0BF8A	21C0C094	23534704	23534822	266	286	-10	
24	2021-05-26 15:10:44	257298BE	257299C8	2705A3FC	2705A512	266	278	-6	
25	2021-05-26 15:10:44	2926A976	2926AA8E	2AB6DFB2	2AB6E0C2	280	272	4	
26	2021-05-26 15:10:44	2CD72CBE	2CD72DD2	2E68C74E	2E68C85E	276	272	2	
27	2021-05-26 15:10:44	3088EF7A	3088F092	321B64B2	321B65C6	280	276	2	
28	2021-05-26 15:10:44	343BB592	343BB6AE	35CCE02E	35CCE13A	284	268	8	
29	2021-05-26 15:10:44	37EDD252	37EDD366	397F9C26	397F9D36	276	272	2	
30	2021-05-26 15:10:45	0004629A	000463B2	0198AEF2	0198B002	280	272	4	

Method 2 - From PTP Debug File to Log

```
Raw Input: Tx 000060ADFF2B.0146EDAA, Rx 000060ADFF2B.0146EF94
Ts diff (with Corr sec): Tx 60ADFF2B.0146EDAA, Rx 60ADFF2B.0146EF94, CorrSec 00000000, stream:1
Raw Input: Tx 000060ADFF2B.016D3710, Rx 000060ADFF2B.016D3904
Ts diff (with Corr sec): Tx 60ADFF2B.016D3710, Rx 60ADFF2B.016D3904, CorrSec 00000000, stream:0
Raw Input: Tx 000060ADFF2B.04F8F5A0, Rx 000060ADFF2B.04F8F794
Ts diff (with Corr sec): Tx 60ADFF2B.04F8F5A0, Rx 60ADFF2B.04F8F794, CorrSec 00000000, stream:1
Raw Input: Tx 000060ADFF2B.0526E3B0, Rx 000060ADFF2B.0526E59B
Ts diff (with Corr sec): Tx 60ADFF2B.0526E3B0, Rx 60ADFF2B.0526E59B, CorrSec 00000000, stream:0
Raw Input: Tx 000060ADFF2B.08AAB354, Rx 000060ADFF2B.08AAB164
Ts diff (with Corr sec): Tx 60ADFF2B.08AAB354, Rx 60ADFF2B.08AAB164, CorrSec 00000000, stream:1
```

Hex Calculator

Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
0146EF94 - 0146EDAA = **1EA**

Decimal value:
21426068 - 21425578 = **490**

0146EF94 - 0146EDAA = ?

Calculate Clear

```
RP/0/RP0/CPU0:May 26 16:56:27.310 KST: ptp_ctrlr[1159]: $PLATFORM-PTP-6-SERVO_EVENTS : PTP Servo state transition from state PHASE_LOCKED to state FREQ_LOCKED
RP/0/RP0/CPU0:May 26 16:56:57.816 KST: ptp_ctrlr[1159]: $PLATFORM-PTP-6-SERVO_EVENTS : PTP Servo state transition from state FREQ_LOCKED to state PHASE_LOCKED
```

2 approaches to calculate a bad timestamp

	Method 1	Method 2
Approach	From show log to PTP Debug File	From PTP Debug File to show log
Advantage	Light calculate Workload	Can find all bad timestamp
Disadvantage	May be not find all bad timestamp	Heavy calculate workload
Assumption	PTP Timestamp matches with log	N/A

A perfect solution for calculating a PTP Debug file

- All People can use the solution.
- Easy to use.
- Calculate all offset timestamp for PTP Debug file
- Show bad timestamps.
- Report for the result of PTP Debug
- The performance of solution should acceptable

Solution for calculating a PTP Debug file

PTP Badtimestamp Analyzer Program	
Run PTP Badtimestamp Analyzer	
PTP Badtimestamp Analyzer Report	
PTP Badtimestamp Analyzer Statistics	
PTP Debug File Name PTP Debug File Size PTP Badtimestamp Analyzer Start Time~End Time PTP Badtimestamp Analyzer Running Time	
PTP Debug Log Statistics	
Total Log Count TotalSync Count No Steam0 Count Badtimestamp Count	
PTP Debug Log Analysis	
PTP Debug Log Start Time ~ End time PTP Debugging time	

PTP Bad timestamp Analyzer(PBA)



How to run a PBA

Enable active content

1

2

PTP Badtimestamp Analyzer Program

Run PTP Badtimestamp Analyzer

PTP Badtimestamp Analyzer Report

PTP Badtimestamp Analyzer Statistics

PTP Debug File Name
PTP Debug File Size
PTP Badtimestamp Analyzer Start Time~End Time
PTP Badtimestamp Analyzer Running Time

Choose PTP Debug file

3

Please Choose PTP Debug File

Name	Date modified	Type	Size
iCON	6/4/2021 9:26 AM	File folder	
GitHub Desktop	5/13/2021 8:27 AM	Shortcut	3 KB
myBase Desktop 7	12/14/2020 8:43 PM	Shortcut	2 KB
ptp.png	6/13/2021 10:13 AM	PNG File	115 KB
ptp_trace_file.log	6/11/2021 4:10 PM	UltraEdit Docume...	2,203 KB
ptp_trace_file_IVR-C01_text.log	6/10/2021 10:26 AM	UltraEdit Docume...	429,759 KB
ptp_trace_file_IVR-C01_text1.log	6/11/2021 2:50 PM	UltraEdit Docume...	2 KB
ptp_trace_file_IVR-C01_text122.log	6/10/2021 10:26 AM	UltraEdit Docume...	429,759 KB
router_conf.txt	3/23/2021 4:16 PM	UltraEdit Docume...	3 KB
test.xlsm	6/10/2021 4:18 PM	Microsoft Excel M...	27 KB
Visual Studio Code	11/26/2020 1:26 PM	Shortcut	2 KB
有道云笔记	2/3/2021 1:48 PM	Shortcut	2 KB
网易有道词典	11/30/2020 12:22 PM	Shortcut	2 KB

How to run a PBA

Review PTP_Debug file

PTP Badtimestamp Analyzer Program

Run PTP Badtimestamp Analyzer

PTP Badtimestamp Analyzer Report

PTP Badtimestamp Analyzer

PTP Debug File Name
PTP Debug File Size
PTP Badtimestamp Analyzer Start Time~End Time
PTP Badtimestamp Analyzer Running Time

Total Log Count
TotalSync Count
No Steam0 Count
Badtimestamp Count

PTP Debug Log Analysis

PTP Debug Log Start Time ~ End time
PTP Debugging time

PTP Debug File :
C:\ptp_trace_file.log

Please confirm to analysis the log

OK Cancel

4

Review PBA Report

PTP Badtimestamp Analyzer

PTP Badtimestamp Analyzer Final Report is completed
Please review PTP Badtimestamp Analyzer Report

OK

5

PTP Badtimestamp Analyzer Report

PTP Badtimestamp Analyzer Statistics

PTP Debug File Name	D:\005.Download\ptp_trace_file_JVR-C01.log
PTP Debug File Size	419.69 MB
PTP Badtimestamp Analyzer Start Time~End Time	2021-06-18 13:20:29~2021-06-18 13:38:35
PTP Badtimestamp Analyzer Running Time	0:18:06

PTP Debug Log Statistics

Total Log Count	1,122,920
TotalSync Count	1,121,628
No Steam0 Count	1,292
Badtimestamp Count	1

PTP Debug Log Analysis

PTP Debug Log Start Time ~ End time	2021-05-26 15:10:43~2021-05-27 10:31:12
PTP Debugging time	19:20:29

Check Bad timestamps in data file

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	PTP Badtimestamp Analyzer Program													
2														
3														
4														
5														
6														
7	Run PTP Badtimestamp Analyzer													
8														
9														
10														
11	PTP Badtimestamp Analyzer Report													
12														
13														
14														
15	PTP Badtimestamp Analyzer Statistics													
16														
17	PTP Debug File Name		D:\005.Download\ptp_trace_file_IVR-C01.log											
18	PTP Debug File Size		419.69 MB											
19	PTP Badtimestamp Analyzer Start Time~End Time		2021-06-18 13:20:29~2021-06-18 13:38:35											
20	PTP Badtimestamp Analyzer Running Time		0:18:06											
21	PTP Debug Log Statistics													
22	Total Log Count		1,122,920											
23	TotalSync Count		1,121,628											
24	No Steam0 Count		1,292											
25	Badtimestamp Count		1											
26														
27	PTP Debug Log Analysis													
28	PTP Debug Log Start Time ~ End time		2021-05-26 15:10:43~2021-05-27 10:31:12											
29	PTP Debugging time		19:20:29											
30														
31														
32														
33														
34														
35														

Please feel free to contact me if you need help.
Maintained by guangli2@ciseco.com

Main PTP_TS_2 PTP_TS_1

	A	B	C	D	E	F	G	H	I	J
1	Debug_Time	T1	T2	T3	T4	T2-T1	T4-T1	Offset	BTSYI	
102427	2021-05-26 16:56:34	016D5CD8	016D38F7	01A0E364	01A0E554	-9185	496	-4840.5 Y		
1000002										
1000003										
1000004										
1000005										
1000006										
1000007										
1000008										
1000009										
1000010										
1000011										
1000012										
1000013										
1000014										
1000015										
1000016										
1000017										
1000018										
1000019										
1000020										
1000021										
1000022										
1000023										

Performance



PBA processes big file size faster than small file size/sec

File Info	Test case 1	Test case 2	Test case 3
File Name	ptp_trace_file_IVR-C01.log	ptp_trace_file_180.log	ptp_trace_file.log
File Size	420MB	179 MB	2MB
Log Count	1,121,628	405,488	4,905
Using Time(sec)	952s(16min)	412s(7min)	6s
Process Sync/second	1178	984	817

Limitation



Limitation

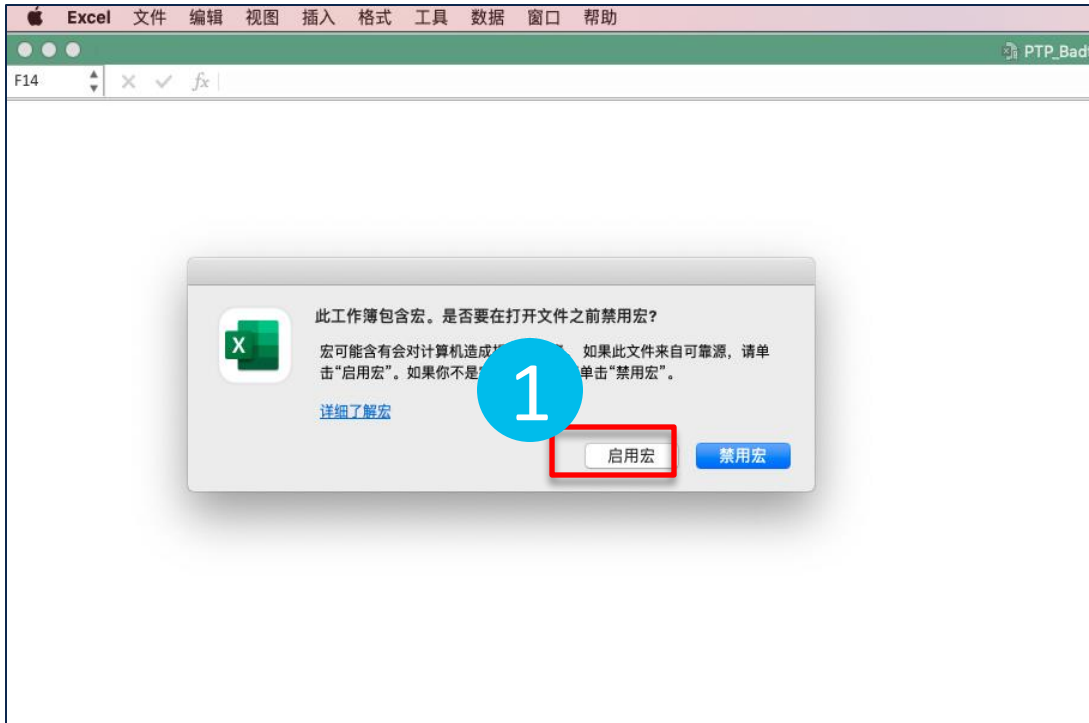
- The Biggest file Size for test is **595MB** in test.
There is no test for file size over 595MB.
- Put PTP Debug File in **English Name** Folder.
No Korean and Chinese Name Folder
- **Close the PTP Debug File** which will be processed.
- **Don't run the other excel files**, when the PBA runs.
- When PBA processes big size file, it would be stuck.
Drink a coffee and take a rest until PBA completion.

PBA Mac Version Released!!!



How to run PBA Mac Version

Similar operation like Microsoft Version



The screenshot shows the PTP Badtimestamp Analyzer Program interface. The top yellow bar contains the program name "PTP Badtimestamp Analyzer Program" (labeled 2) and a button "Run PTP Badtimestamp Analyzer" (labeled 3). Below this is a section "Please input file path" with a text input field. The main section is titled "PTP Badtimestamp Analyzer Report". It is divided into three sub-sections: "PTP Badtimestamp Analyzer Statistics" (containing "PTP Debug File Name", "PTP Badtimestamp Analyzer Start Time~End Time", and "PTP Badtimestamp Analyzer Running Time"), "PTP Debug Log Statistics" (containing "Total Log Count", "TotalSync Count", "No Steam0 Count", and "Badtimestamp Count"), and "PTP Debug Log Analysis" (containing "PTP Debug Log Start Time ~ End time" and "PTP Debugging time").

Microsoft VS Mac

- No PTP Debug File size
- Input file path

PTP Badtimestamp Analyzer Program	
<div>Run PTP Badtimestamp Analyzer</div>	
PTP Badtimestamp Analyzer Report	
PTP Badtimestamp Analyzer Statistics	
PTP Debug File Name PTP Debug File Size PTP Badtimestamp Analyzer Start Time~End Time PTP Badtimestamp Analyzer Running Time	
PTP Debug Log Statistics	
Total Log Count TotalSync Count No Steam0 Count Badtimestamp Count	
PTP Debug Log Analysis	
PTP Debug Log Start Time ~ End time PTP Debugging time	

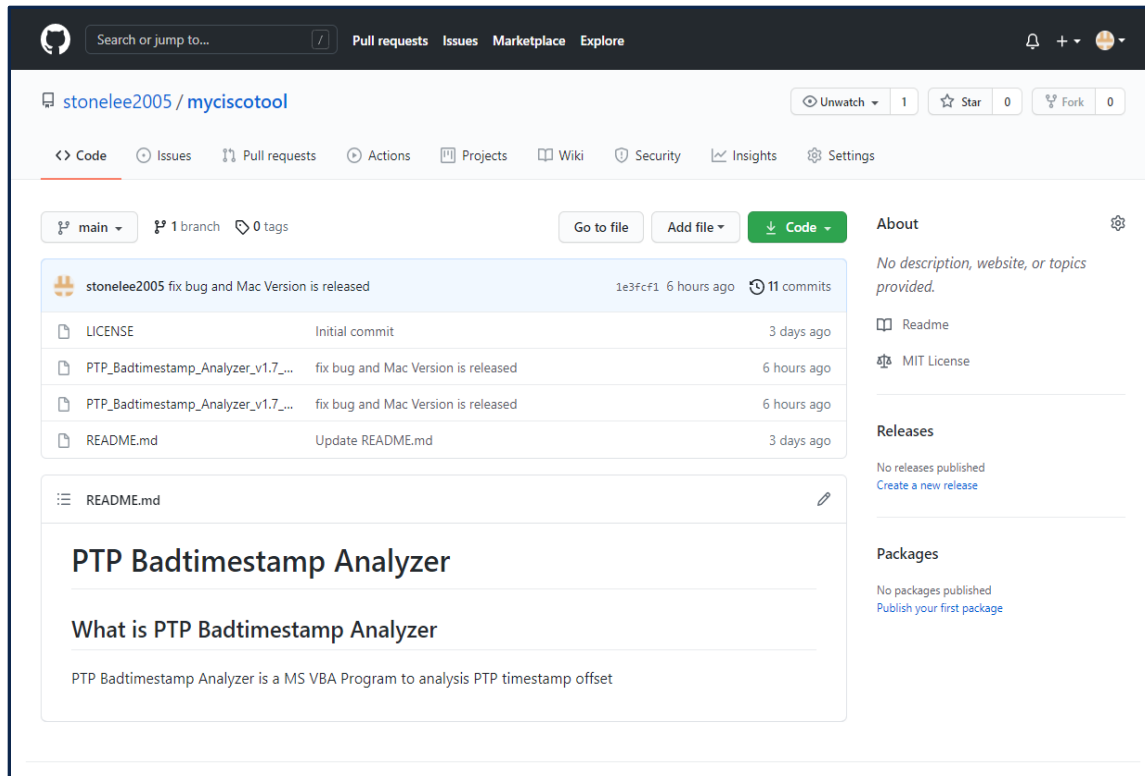
PTP Badtimestamp Analyzer Program	
<div>Please input file path <input type="text"/></div>	<div>Run PTP Badtimestamp Analyzer</div>
PTP Badtimestamp Analyzer Report	
PTP Badtimestamp Analyzer Statistics	
PTP Debug File Name PTP Badtimestamp Analyzer Start Time~End Time PTP Badtimestamp Analyzer Running Time	
PTP Debug Log Statistics	
Total Log Count TotalSync Count No Steam0 Count Badtimestamp Count	
PTP Debug Log Analysis	
PTP Debug Log Start Time ~ End time PTP Debugging time	

How to get a PBA



Get PBA from Github

■ Get PBA from <https://github.com/stonelee2005/myciscotool>



stonelee2005 / myciscotool

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

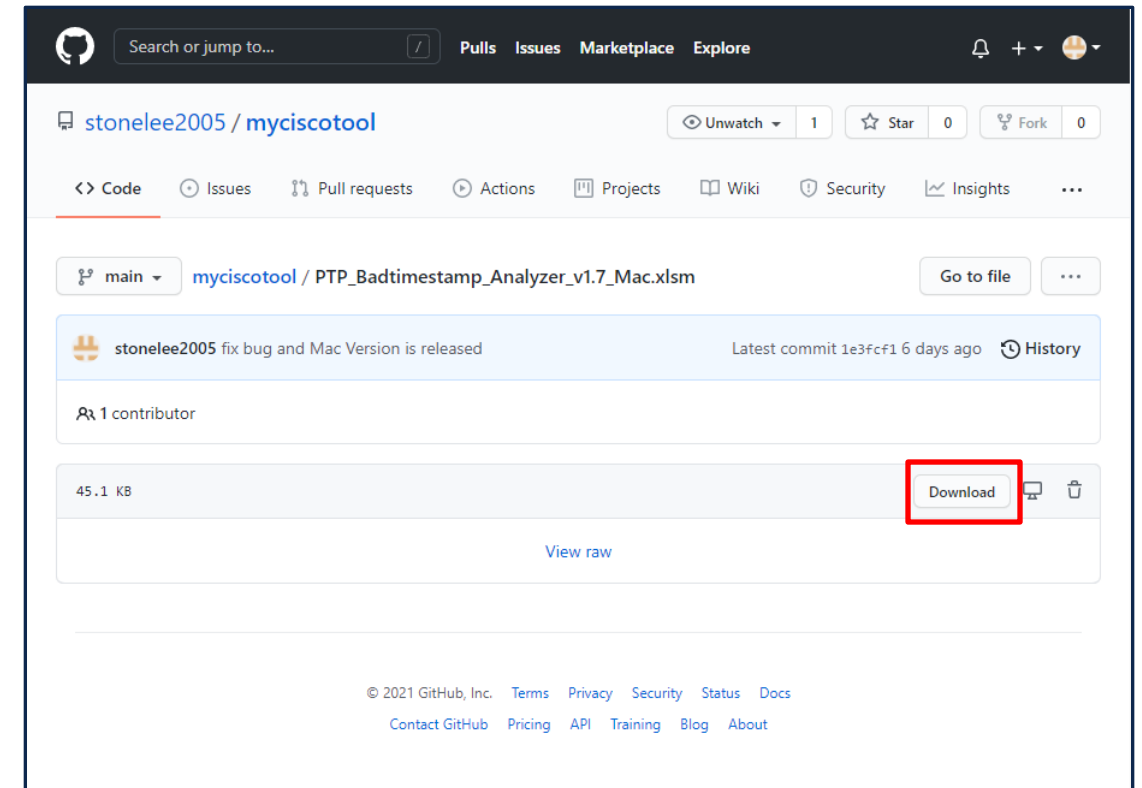
stonelee2005 fix bug and Mac Version is released 1e3fcf1 6 hours ago 11 commits

File	Commit	Time
LICENSE	Initial commit	3 days ago
PTP_Badtimestamp_Analyzer_v1.7_...	fix bug and Mac Version is released	6 hours ago
PTP_Badtimestamp_Analyzer_v1.7_...	fix bug and Mac Version is released	6 hours ago
README.md	Update README.md	3 days ago

PTP Badtimestamp Analyzer

What is PTP Badtimestamp Analyzer

PTP Badtimestamp Analyzer is a MS VBA Program to analysis PTP timestamp offset



stonelee2005 / myciscotool

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights ...

main myciscotool / PTP_Badtimestamp_Analyzer_v1.7_Mac.xlsm

Go to file ...

stonelee2005 fix bug and Mac Version is released Latest commit 1e3fcf1 6 days ago History

1 contributor

45.1 KB

Download

View raw

© 2021 GitHub, Inc. Terms Privacy Security Status Docs
Contact GitHub Pricing API Training Blog About

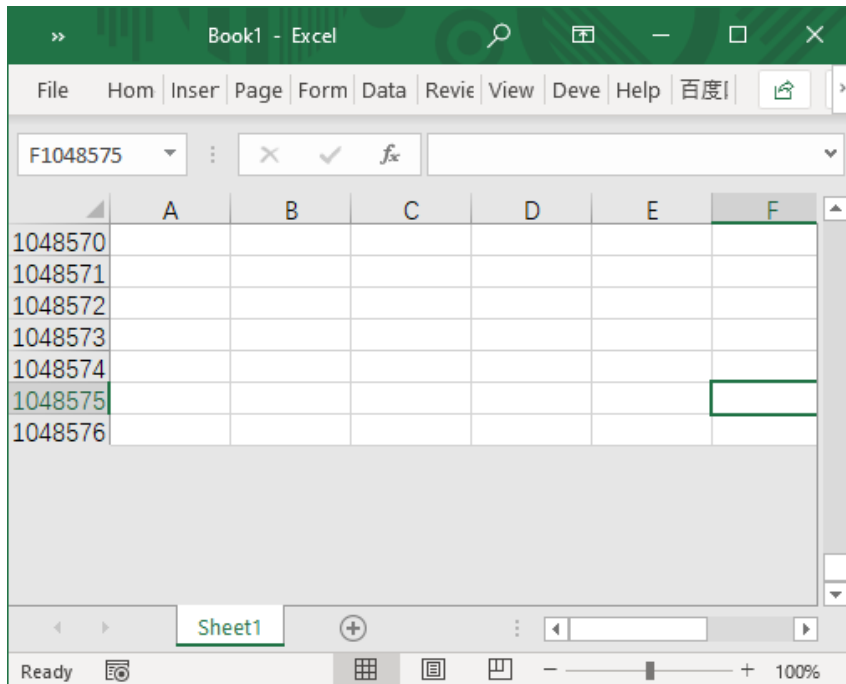
Q & A



Question 1

1,048,576 row limitation per 1 sheet in MS Excel.

If PBA data can process row over 1,048,576 ?



Answer 1

PBA will create a new sheet if row over 1,000,000.

	A	B	C	D	E	F	G	H	I	J
1	Debug_Time	T1	T2	T3	T4	T2-T	T4-T	Offset	BTSY	
102427	2021-05-26 16:56:34	016D5CD8	016D38F7	01A0E364	01A0E554	-9185	496	-4840.5	Y	
1000002										
1000003										
1000004										
1000005										
1000006										
1000007										
1000008										
1000009										
1000010										
1000011										
1000012										
1000013										
1000014										
1000015										
1000016										
1000017										
1000018										
1000019										
1000020										
1000021										
1000022										
1000023										

	A	B	C	D	E	F	G	H	I	J	K
1	Debug_Time	T1	T2	T3	T4	T2-T	T4-T	Offset	BTSY		
122922											
122923											
122924											
122925											
122926											
122927											
122928											
122929											
122930											
122931											
122932											
122933											
122934											
122935											
122936											
122937											
122938											
122939											
122940											
122941											
122942											
122943											
122944											

Question 2

When close PBA, data will be saved?

Answer 2

PBA clears PBA Report and data Sheet, When close PBA.

Run PTP Badtimestamp Analyzer

1 PTP Badtimestamp Analyzer Report

PTP Badtimestamp Analyzer Statistics

PTP Debug File Name
PTP Debug File Size
PTP Badtimestamp Analyzer Start Time~End Time
PTP Badtimestamp Analyzer Running Time

PTP Debug Log Statistics

Total Log Count
TotalSync Count
No Steam0 Count
Badtimestamp Count

PTP Debug Log Analysis

PTP Debug Log Start Time ~ End time
PTP Debugging time

Please feel free to contact me if you need help.
Maintained by guangli2@cisco.com

2

Main

Microsoft Excel
Want to save your changes to 'PTP_Badtimestamp_Analyzer_v1.7.xlsm'?
Save Don't Save Cancel



The bridge to possible

Thank you

