Test Guide
CX5036 v7.4

Enhanced Noise Transfer Script
Getting Started Guide

ITU-T G.8273.2 Boundary Clock Testing

- Noise Transfer

This Application Note describes how to use the Calnex Paragon-X to prove T-BC compliance to G.8273.2 Clause 7.3, Noise Transfer.

Table of Contents

1 Introduction

To allow users to test noise transfer in line with ITU-T G.8273.2 (Clause 7.3), Calnex has created an Enhanced Noise Transfer (ENT) tool for Paragon-X. This tool come in 2 forms

- Tcl script
- Python script

Noise Transfer Overview

Time error transfer is a measure of how much “noise” (time error) present at the input is transferred to the output of the clock. The clock should act as a filter, removing some of the noise present at the input, therefore it is usually expressed in terms of a filter bandwidth.

There are two filter characteristics described in the G.8273.2 specification. The first is the time error transfer from PTP to PTP (and also PTP to 1pps), i.e. the boundary clock itself. PTP interfaces are generally quite noisy at frequencies close to the message rate. This is because of quantization noise in the timestamps, and also positional quantization in the physical layer components as the packets cross from the line clock domain to the internal clock domain. Therefore, G.8273.2 defines a low-pass filter function from PTP input to PTP output to remove the interface noise. The bandwidth of this filter is between 0.05 and 0.1Hz, as shown in Figure 1:

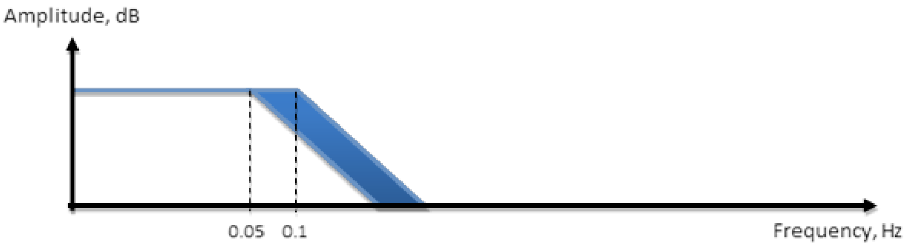


Figure 1: PTP to PTP Transfer Function

The transfer function from the SyncE input to PTP output is more complex. Firstly, there is a low-pass function in the EEC itself, defined in G.8262 to have a bandwidth of between 1 and 10Hz. The boundary clock then acts as a high-pass filter to the SyncE signal, at the same bandwidth as the PTP to PTP low pass filter. This high-pass function as a natural consequence of the way the clock works: it smooths out high frequency noise by following the local frequency reference (in this case, the SyncE), while following the PTP input at low frequencies.

The net result is that any noise (or phase wander) on the SyncE input is band-pass filtered, as shown in Figure 2:

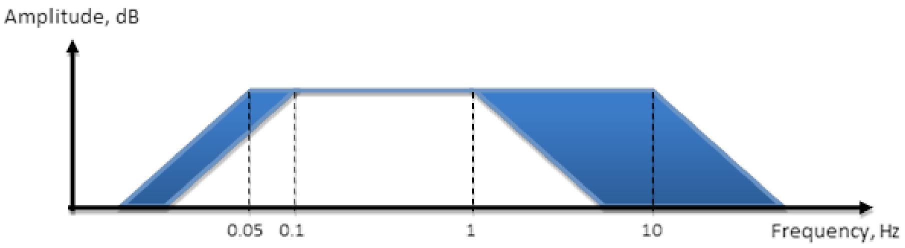


Figure 2: SyncE to PTP Transfer Function

The purpose of noise transfer testing is to determine if the filter characteristic is within the limits defined in the standard, i.e. the bandwidth is correct, any gain in the pass-band is limited, and there is sufficient attenuation in the stop-band. This is done by varying the input phase with a series of sinusoidal tones of different frequencies, and measuring the amplitude of these tones at the output.

Further detailed explanations of the theory of measuring Time Error Transfer and how this applies to the testing of Boundary Clocks can be found in the Calnex companion documents to this document available to download from the Calnex Website

https://www.calnexsol.com/en/resources-en/technology-library

- [CX5034] Measuring Time Error Transfer of G.8273.2 T-BCs
- [CX5033] Testing a T-BC to ITU-T G.8273.2

Script Overview

The script measures the filter response of a T-BC in the following scenarios:

- PTP to PTP
- SyncE to PTP
- PTP to 1PPS
- SyncE to 1PPS

The noise applied to the T-BC input is as suggested in G.8273.2 Appendix VI. The script (with associated profiles) controls the Paragon-X to apply this noise and then measures the response of the DUT.

G.8273.2 Appendix VI also suggests that a filter is applied to the measured output to address potential measurement uncertainty due to noise on the associated interface and the intrinsic noise generation of the T-BC. The script implements the filter using least-squares estimation (one of the filter techniques listed in Appendix VI).

The script provides a pass / fail result – again based on the limits suggested in G.8273.2 Appendix VI.

Notes:

- G.8273.2 specifies that testing is performed in the context of G.8275.1; this uses a forward and reverse message rate of 16 packets/sec.
- The frequencies suggested in G.8273.2 Appendix VI are carefully chosen to avoid Nyquist effects at this message rate.
- This means that the script results will be valid ONLY for 16 packets/second and using the defined frequencies. These parameters in the script should not be modified.

PTP to PTP and PTP to 1PPS Noise Transfer

Table 1 below shows the stimulus frequencies, amplitudes and durations of the noise applied to the PTP input of the DUT. Note that 200ns of sinusoidal time error is achieved by applying a sinusoidally varying delay in one direction with an amplitude of 400ns (the overall time error is an average of the error in both directions). The table lists the permitted gains and output amplitudes based on a noise-free T-BC (clean) and with an allowance for an additional level of noise on the output. The allowance (N) used in the script is 10ns.

Point	**Test Freq (Hz)**	**Pk-Pk I/P Ampl (ns)**	**Pk-Pk O/P Ampl ±*10* ns added noise allowance (ns)**		**Duration (s)**
			Max	Min	
0	0.00390625	200	215	130	800
1	0.0078125		215	130	
2	0.015625		215	130	400
3	0.03125		215	130	250
4	0.0615625		215	n/a	
5	0.123125		140		
6	0.24625		90		
7	0.4925		50		
8	0.985		35		
9	1.985		30		

Table 1: Maximum and Minimum Expected Output Amplitudes at Test Frequencies for PTP-to-PTP and PTP-1PPS Noise Transfer Measurement

SyncE to PTP and SyncE to 1PPS Noise Transfer

Table 2 shows the stimulus frequencies, amplitudes and durations of the noise applied to the SyncE input of the DUT. The table lists the permitted gains and output amplitudes based on a noise-free T-BC (clean) and with an allowance for an additional level of noise on the output. The allowance (N) used in the script is 25ns.

Point	**Test Freq (Hz)**	**Pk-Pk I/P Ampl (ns)**	**Pk-Pk O/P Ampl ±25 ns added noise allowance (ns)**		**Duration (s)**
			Max	Min	
0	0.00390625	2000	225	n/a	800
1	0.0078125		340		
2	0.015625		630		400
3	0.03125	1000	565		250
4	0.0615625	500	545		
5	0.123125	250	285	150	
6	0.24625				
7	0.4925				
8	0.985				
9	1.985			n/a	
10	3.985				
11	7.985				

Table 2 Maximum and Minimum Expected Output Amplitudes at Test Frequencies for SyncE-to-PTP and SyncE-to-1pps Noise Transfer Measurement

2 Using the Enhanced Noise Transfer Script

Overview

The purpose of this guide is to familiarise users with the features and variables of the script. Basic knowledge of Tcl/Python scripting is assumed throughout.

The script performs the noise transfer test by loading an impairment profile (one per frequency) into the Paragon-X and then replaying the file onto either the SyncE or PTP output on port 1 of the Paragon-X i.e. providing a noisy input to the DUT. The output of the DUT (1pps and/or PTP) is then measured on port 2.

The output results are filtered using a Least Squares estimation algorithm (as suggested in G.8273.2 Amendment 1). The filtered results are then used to calculate gain and are also compared to the limits defined above.

The final results are output as csv files. If the optional UI is enabled, the results are also stored as a screen capture of the UI.

System Configuration

The script can be run on the PC hosting the Paragon-X Application (***Local Host***) or can be run from another PC. When used from another PC, the computer running the script is not required to be Windows-based – any computer capable of running Tcl or Python can be used.

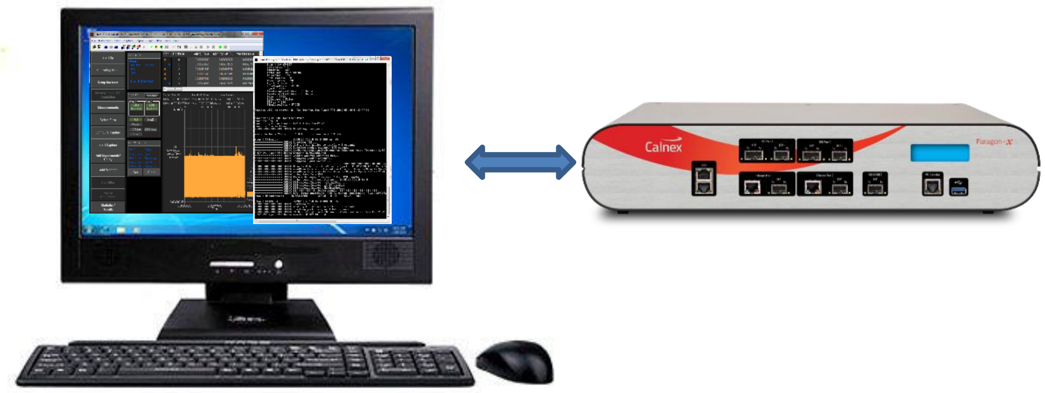


Figure 3: PC *(Local Host)* running Paragon-X Application and the ENT script

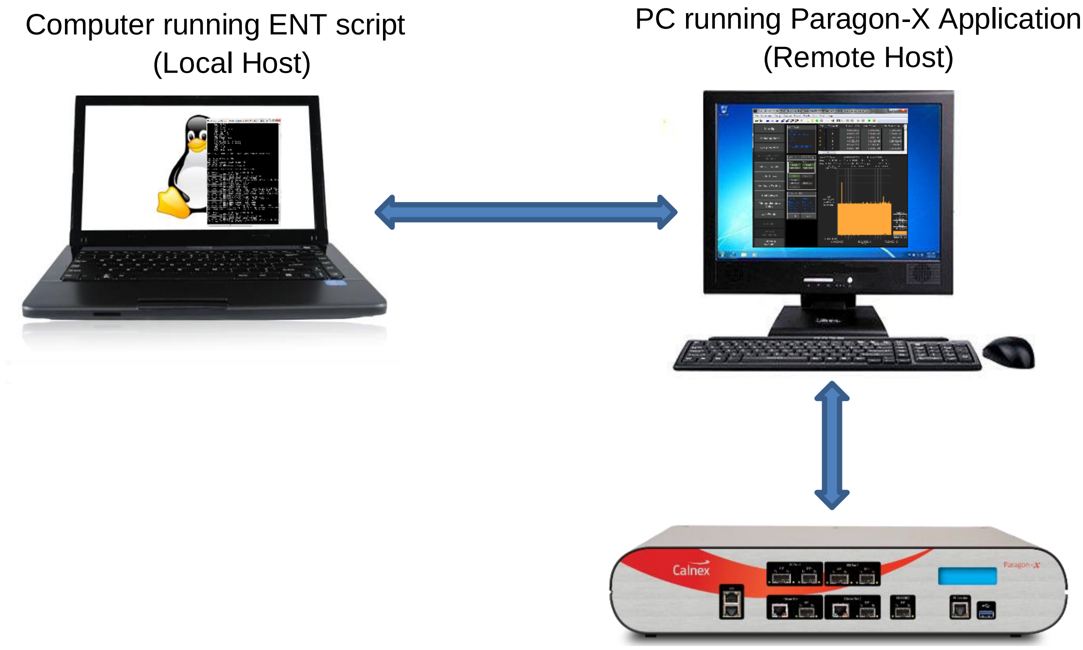


Figure 4: Remote PC running Paragon-X Application

Note that when connecting to a remote PC:

- The location of files is split between the local host and the remote Windows PC. The file locations used for remote and local use are different – these differences are noted in the sections below.
- This guide refers to the Windows PC hosting the Paragon-X application as the *remote host* and the computer hosting the ENT script as the *local host*.

Most of this document assumes that the script and the Paragon-X Application are running on the same PC e.g. on a *local host*. Section 4 describes how to use the script remotely.

TCL

System Requirements

Running the TCL test script requires Tcl version 8.6 or higher, or Tcl 8.5.8.1 with OO extensions installed.

If the script is run with Graphing enabled, then the *Tk*, *Img* and *rbc* libraries are also required. The script will detect if these packages are not present and will notify the user of any that are missing. To install the libraries use a Tcl package installer tool like ‘teacup.exe’ (installed as part of ActiveState ActiveTcl)

```
teacup.exe install img
```

The PC requirements for the Paragon-X and associated tools are described in “Minimum PC Requirements” installed with the Paragon-X Application.

To use the ENT script to connect to a remote Windows PC (see Section 4), the Windows PC must be accessible to the *local host* and port 1337 must be open. In addition, the *REST* Tcl library (version 1.0.1 or later) is required. This can be installed by using a TCL package installer like ‘teacup.exe’ (installed with ActiveState ActiveTcl).

```
teacup install rest
```

Installation

The script and associated files are delivered as a zip file.The contents of the zip file should be extracted to a folder of your choice. No further installation is required.

Configuration

The configurable parameters can be controlled from a combination of command line arguments, the *Defaults.tcl* file or from a specified user configuration file (see Executing the Enhanced Noise Transfer Test for more details).

It is also possible to modify or add settings directly in the scripts. Such modifications or additions may result in incorrect operation and so should be undertaken only when the standard script cannot be used directly.

The configurable parameters control the behaviour of the Paragon-X or the behaviour of the script itself. These parameters are listed in the tables below. Note that the parameters and settings are case-sensitive.

Setting	Description	Default	Allowed Values
`InstrIp`	The IP address of the Paragon-X management port.	`192.168.3.100`	Valid IP address
`GuiIp`	The IP address of the PC hosting the Paragon-X application	`localhost`	Valid IP address (see Note 1)
`GuiPort`	The TCP port for script to application software commands	`9000`	Valid TCP port (see Note 2)
`LineRate`	Line rate to be used by both Paragon-X ports.	`1GBE`	`100BASET, 1GBE, 10GBE (see Note 3)`
`Interface`	Interfaceto be used for both Paragon-X ports	`ELECTRICAL`	`XFP, SFPPLUS, ELECTRICAL, OPTICAL (See Note 3)`
`AutoNegotiate`	Enables/disables auto negotiation for both Paragon-X ports	`FALSE`	`TRUE, FALSE (see Note 3)`
`Port1Config`	Independent port 1 configuration parameters	`(see Note 4)`	`(see Note 4)`
`Port2Config`	Independent port 2 configuration parameters	`(see Note 4)`	`(see Note 4)`
`TwoStep`	Enables/disables two-step operation	`FALSE`	`TRUE, FALSE`
`SyncE`	Enable/disable SyncE measurement during the test	`TRUE`	`TRUE, FALSE`
`Reference`	Source of the Paragon-X frequency reference.	`INT`	`EXT_10M,EXT_R75_E1, EXT_R100_T1,EXT_R100_E1, EXT_K64, INT`

Setting	Description	Default	Allowed Values
`CableDelay`	Cable delay between the DUT and Paragon-X port 2. (see Note 5)	`5`	Type: Integer Range: 0 to 1000 Unit: ns
`1ppsThreshold`	When `Type` = `PTP-1PPS` or `SyncE-1PPS`	`1.0`	Type: Float Range: 0.5 to 2.5 Units: Volts
`1ppsImpedance`	Impedance level for Upper Aux Measurement Port	`R50`	R50, HIGH_IMPEDANCE (see Note)

Table 3: Instrument Settings

Notes:

1. **`GuiIp:`** If the IP address of the GUI (Paragon-X application) is not “localhost”, then it is assumed that the Paragon-X application is running on a remote host. This will change the behaviour of the script and may also change the locations of saved files (see Using the Script Remotely).
2. **`GuiPort:`** The GUI (Paragon-X application) listens on this port for commands from active scripts. By default port 9000 is used by all instances of the GUI. You can override this for individual GUI instances using a command line option when launching the GUI. See the Paragon-X Remote Control Reference Manual for details
3. **`LineRate/Interface/AutoNegotiate:`** These parameters will be applied to both Paragon-X ports, and are only active when the **Port1Config** and **Port2Config** parameters are not present.
4. **`Port1Config/Port2Config:`** These parameters enable independent configuration of Paragon-X port 1 and port 2. All three parameters **LineRate, Interface** and **AutoNegotiate** must be stated for **Port1Config** and **Port2Config.** An error will be raised if any are missing, or if only Port1Config or Port2Config is present. Allowed parameter values are the sames as the above and there are no default values.

   `Parameter Format:`

   Example: Port1Config “1GBE ELECTRICAL TRUE” Port2Config “10GBE OPTICAL FALSE”

5. **`CableDelay:`** This is not required since noise transfer measures peak-to-peak output and this is not affected by cable delay. However, if used, for electrical interfaces, the delay due to the cable will be approximately 5.1 ns per metre; for optical fibre, the delay will be approximately 4.9 ns per metre. Only full nanosecond values are counted, so calculated values need to be rounded to the nearest full nanosecond value.
6. **`1ppsImpedance:`** This is only applicable to **revision 3** of the Paragon-X. In previous revisions changes to this will have no effect. To determine revision select the ***Instrument ->Details*** menu item in the Application user interface.

Setting	Description	Default	Allowed Values
Type	Test type to be performed (see note 1)	PTP-BOTH	PTP-BOTH,PTP-PTP, PTP-1PPS,SyncE-BOTH, SyncE-PTP,SyncE-1PPS
PtpPoints	Specifies which frequency points are to be tested with PTP input	ALL	ALL or list of points (see note 2)
SyncEPoints	Specifies which frequency points are to be tested with SyncE input	ALL	ALL or list of points (see note 2)
ConfigFile	Specifies a Tcl file containing configuration parameters (see Executing the Enhanced Noise Transfer Test)	“”	Valid Tcl file
SettlingTime	Time for the device to lock to PTP supplied from the P-X Master (see note 3).	120	Type: Integer Range: Positive Unit: Seconds
RecoveryTime	Time for the device to recover after a change to the applied input noise (see note 4).	50	Type: Integer Range: Positive Unit: Seconds
LimitsEnabled	Specifies whether to check measured results against pre-defined limits. Disabling limit checking allows users to use their own limits.	TRUE	TRUE, FALSE
MasterMcastAddress	Specifies whether the Forwardable Ethernet multicast address (01:1b:19:00:00:00) or non Forwardable address (01:80:c2:00:00:0e) should be used by the master	FORWARDABLE	FORWARDABLE, NONFORWARDABLE
SlaveMcastAddress	Specifies whether the Forwardable Ethernet multicast address (01:1b:19:00:00:00) or non Forwardable address (01:80:c2:00:00:0e) should be used by the slave	FORWARDABLE	FORWARDABLE, NONFORWARDABLE
Graphing	Enables/disables UI presentation of results	FALSE	TRUE, FALSE
GraphingYAxis	When **Graphing** is TRUE, plot results as either Gain(dB) or Amplitude(ns)	GAIN	AMPLITUDE, GAIN
Debug	Enables/disables additional debugging output to the console	FALSE	TRUE, FALSE
ReProcess	See “Re-Processing Previously Captured Data” below	FALSE	TRUE, FALSE

Setting	Description	Default	Allowed Values
`RemoteCaptureFolder`	See Using the Script Remotely	`C:/Desktop`	
`RemoteProfileFolder`	See Using the Script Remotely	`C:/Desktop`	
`Filter`	For Calnex use only. Do not change	`LSQ`	NA
`SimulatedData`	For Calnex use only. Do not change	`FALSE`	NA

Table 4: Script Settings

Notes:

1. **`Type:`** Specifies the stimulus (PTP or SyncE) to be applied to the DUT input and the measurements to be performed on the DUT output. The measurements can be PTP, 1pps or BOTH.
2. **`PtpPoints / SyncEPoints:`** A subset of frequencies can be selected using a space delimited string of numbers indexed from 0. For example, to test only 0.0071825Hz and 0.4925 Hz, set `PtpPoints` to `“1 7”`. To test all frequencies (as defined in the tables above), then set `PtpPoints` and `SyncEPoints` to `ALL`.
3. **`SettlingTime:`** This should be long enough for the time error measurement to become stable, otherwise the results will be incorrect. This time will vary between DUTs.
4. **`RecoveryTime:`** A fixed period of time to allow the DUT to recover from each change of noise on the DUT input (i.e. for each frequency point). This time will very between DUTs. This recovery time will not be included in the measurement, but the measurement capture duration does include this time.

The defaults listed in table Table 4Table 5 above are contained in *Defaults.tcl*. This file may be modified by the user – the values in the table above are the defaults defined in the original file in the script install.

Paragon-X is automatically configured for the ITU-T G.8275.1 Phase Profile as part of the script initialization process; please ensure that the DUT is also configured for the G.8275.1 profile.

Executing the Enhanced Noise Transfer Test

Setting Configuration Parameters

The configuration parameters can be set in three different ways:

- Defaults: defined in *Defaults.tcl*
- User-specifed configuration file
- Command line arguments

The precedence order is the reverse of the above i.e. command line arguments over-ride settings in any configuration file which over-ride the defaults.

If no setting is specified in the command line arguments or configuration file, then the default value from *Defaults.tcl* will be used.

Note: The *Defaults.tcl* file may be edited by the user. However, for correct operation, ALL settings must be given a default value. Do not delete settings from this file.

Executing From the Command Line

Launch the Paragon-X application on the PC and run ***cmd.exe*** to launch a shell. Change the working directory (*cd*) to the script location. For example:

```
cd “C:\scripts\Script G.8273.2”
```

The script should now be run by launching a Tcl shell passing the script name with the required arguments as command line parameters. The arguments are passed to the script as a simple list of <argument value> pairs (separated by spaces). Some examples are shown below:

```
tclsh G.8273.2.tcl InstrIp 192.168.4.100 LineRate 1GBE Interface OPTICAL
```

The command line arguments specify the instrument IP address, the line rate and the interface to be used. All other settings are defaults (specified in the *Defaults.tcl* file).

```
tclsh G.8273.2.tcl Type PTP-1PPS Reference EXT_10M ConfigFile “config.tcl”
```

The command line arguments specify the test type, the frequency reference. Other settings are defined in the *config.tcl* file. Any settings not specified on the command line or in the configuration file are defaults (specified in the *Defaults.tcl* file).

Executing From a Test Program

If the script is being integrated into another test program, it may not be convenient to execute a Tcl shell to be able to specify command line parameters. In this case, the *Defaults.tcl* file should be edited to set the required parameters (see The Defaults.tcl File below).

In this case, the G.8273.2 script can be executed from Tcl using `source`:

```
source G.8273.2.tcl
```

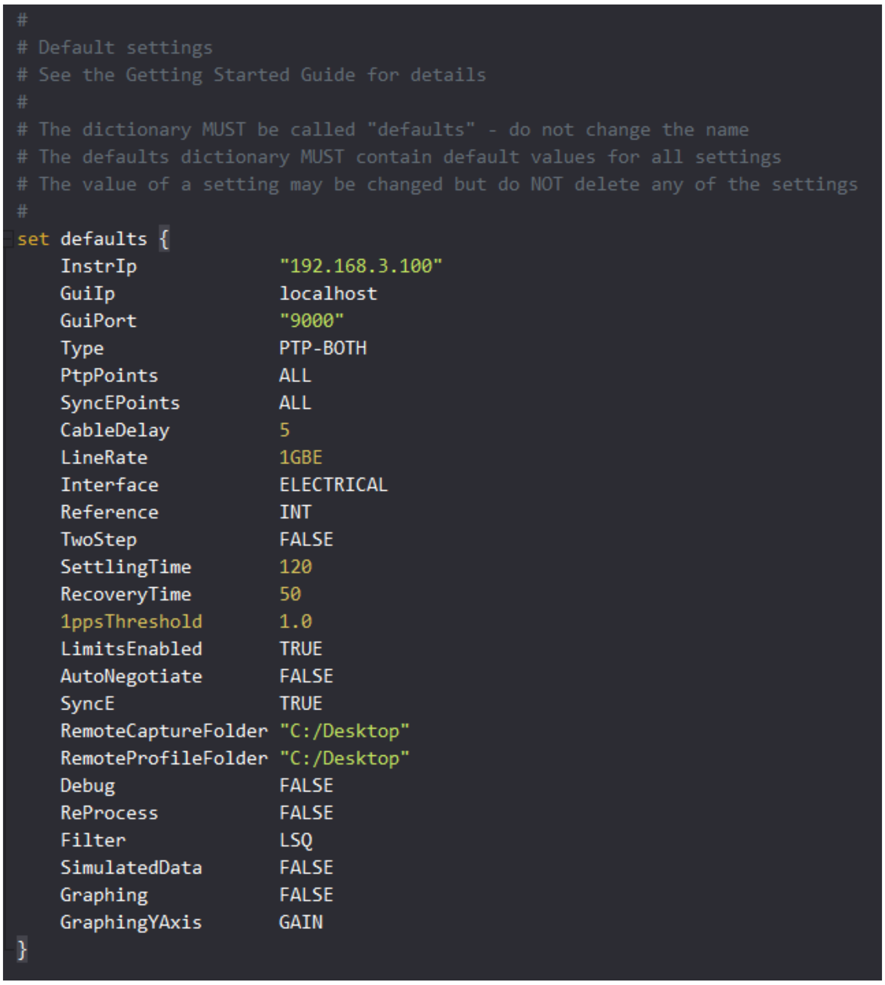
No arguments are required since all settings are defined in the *Defaults.tcl* file.

The Defaults.tcl File

The *Defaults.tcl* file defines the default settings used by the script. This file contains a Tcl dictionary defining the default settings used by the script. This file may be modified by the user if required.

The dictionary in this file must be called “`defaults`”. In addition, the file must define default values for all parameters.

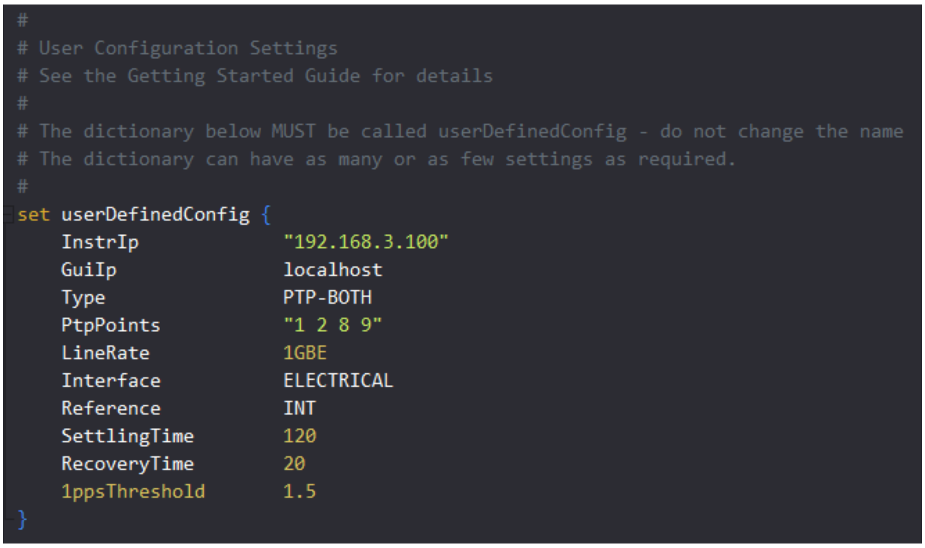
The *Defaults.tcl* file included in the install contains:



```
#
# Default settings
# See the Getting Started Guide for details
#
# The dictionary MUST be called "defaults" - do not change the name
# The defaults dictionary MUST contain default values for all settings
# The value of a setting may be changed but do NOT delete any of the settings
#
set defaults {
    InstrIp             "192.168.3.100"
    GuiIp               localhost
    GuiPort             "9000"
    Type                PTP-BOTH
    PtpPoints           ALL
    SyncEPoints         ALL
    CableDelay          5
    LineRate            1GBE
    Interface           ELECTRICAL
    Reference           INT
    TwoStep             FALSE
    SettlingTime        120
    RecoveryTime        50
    1ppsThreshold       1.0
    LimitsEnabled       TRUE
    AutoNegotiate       FALSE
    SyncE               TRUE
    RemoteCaptureFolder "C:/Desktop"
    RemoteProfileFolder "C:/Desktop"
    Debug               FALSE
    ReProcess           FALSE
    Filter              LSQ
    SimulatedData       FALSE
    Graphing            FALSE
    GraphingYAxis       GAIN
}
```

Configuration Files

A configuration file can also be used to set parameters. This file must contain a Tcl dictionary called “`userDefinedConfig`”. This file can define as many or as few parameters as required. An example configuration file (*ExampleConfig.tcl*) is included in the install. This contains:



```
#
# User Configuration Settings
# See the Getting Started Guide for details
#
# The dictionary below MUST be called userDefinedConfig - do not change the name
# The dictionary can have as many or as few settings as required.
#
set userDefinedConfig {
    InstrIp             "192.168.3.100"
    GuiIp               localhost
    Type                PTP-BOTH
    PtpPoints           "1 2 8 9"
    LineRate            1GBE
    Interface           ELECTRICAL
    Reference           INT
    SettlingTime        120
    RecoveryTime        20
    1ppsThreshold       1.5
}
```

Python

System Requirements

Running the Python test script requires Python version 3.4 or higher.

If the script is run with Graphing enabled, then the *Tk* and *mathplotlib* libraries are also required.

The PC requirements for the Paragon-X and associated tools are described in "Minimum PC Requirements" installed with the Paragon-X Application.

To use the ENT script to connect to a remote Windows PC (see Section 4), the Windows PC must be accessible to the *local host* and port 1337 must be open.

Installation

The script and associated files are delivered as a zip file.The contents of the zip file should be extracted to a folder of your choice. No further installation is required.

Configuration

The configurable parameters can be controlled from a combination of command line arguments, the *Defaults.py* file or from a specified user configuration file (see Executing the Enhanced Noise Transfer Test for more details).

It is also possible to modify or add settings directly in the scripts. Such modifications or additions may result in incorrect operation and so should be undertaken only when the standard script cannot be used directly.

The configurable parameters control the behaviour of the Paragon-X or the behaviour of the script itself. These parameters are listed in the tables below. Note that the parameters and settings are case-sensitive and should be enclosed within " characters.

Setting	Description	Default	Allowed Values
`--InstrIp`	The IP address of the Paragon-X management port.	`192.168.3.100`	Valid IP address
`--GuiIp`	The IP address of the PC hosting the Paragon-X application	`localhost`	Valid IP address (see Note 1)
`--GuiPort`	The TCP port for script to application software commands	`9000`	Valid TCP port (see Note 2)
`--LineRate`	Line rate to be used for both Paragon-X ports	`1GBE`	`100BASET, 1GBE, 10GBE` (see Note 2)
`--Interface`	Interface to be used fo both Paragon-X ports	`ELECTRICAL`	`XFP, SFPPLUS, ELECTRICAL, OPTICAL` (see Note 2)
`--AutoNegotiate`	Enables/disables auto negotiation for both Paragon-X ports	`FALSE`	`TRUE, FALSE`

Setting	Description	Default	Allowed Values
--Port1Config	Independent port 1 configuration parameters	`(see Note 4 )`	`(see Note 4 )`
--Port2Config	Independent port 2 configuration parameters	`(see Note 4 )`	`(see Note 4 )`
--TwoStep	Enables/disables two-step operation	`FALSE`	`TRUE, FALSE`
--SyncE	Enable/disable SyncE measurement during the test	`TRUE`	`TRUE, FALSE`

--Reference	Source of the Paragon-X frequency reference.	`INT`	`EXT_10M,EXT_R75_E1,`
			`EXT_R100_T1,EXT_R100_E1,`
			`EXT_K64, INT`

--CableDelay	Cable delay between the DUT and Paragon-X port 2. (see Note 5)	`5`	Type: Integer Range: 0 to 1000 Unit: ns
--OneppsThreshold	When Type = `PTP-1PPS` or `SyncE-1PPS`	`1.0`	Type: Float Range: 0.5 to 2.5 Units: Volts
--OneppsImpedance	Impedance level for Upper Aux Measurement Port	`R50`	R50, HIGH_IMPEDANCE (see Note)

Table 5: Instrument Settings

Notes:

1. **`GuiIp:`** If the IP address of the GUI (Paragon-X application) is not “localhost”, then it is assumed that the Paragon-X application is running on a remote host. This will change the behaviour of the script and may also change the locations of saved files (see Using the Script Remotely).
2. **`GuiPort:`** The GUI (Paragon-X application) listens on this port for commands from active scripts. By default port 9000 is used by all instances of the GUI. You can override this for individual GUI instances using a command line option when launching the GUI. See the Paragon-X Remote Control Reference Manual for details.
3. **`LineRate/Interface/AutoNegotiate:`** These parameters will be applied to both Paragon-X ports, and are only active when the **Port1Config** and **Port2Config** parameters are not present.
4. **`Port1Config/Port2Config:`** These parameters enable independent configuration of Paragon-X port 1 and port 2. All three parameters **LineRate, Interface** and **AutoNegotiate** must be stated for **Port1Config** and **Port2Config.** An error will be raised if any are missing, or if only Port1Config or Port2Config is present. Allowed parameter values are the sames as the above and there are no default values.

   `Parameter Format:`

   --Port1Config “1GBE,ELECTRICAL,TRUE” --Port2Config “10GBE,OPTICAL,FALSE”

5. **`CableDelay:`** This is not required since noise transfer measures peak-to-peak output and this is not affected by cable delay. However, if used, for electrical interfaces, the delay due to the cable will be approximately 5.1 ns per metre; for optical fibre, the delay will be approximately 4.9 ns per metre. Only full nanosecond values are counted, so calculated values need to be rounded to the nearest full nanosecond value.

6. `OneppsImpedance:` This is only applicable to **revision 3** of the Paragon-X. In previous revisions changes to this will have no effect. To determine revision select the ***Instrument ->Details*** menu item in the Application user interface.

Setting	Description	Default	Allowed Values
`--Type`	Test type to be performed (see note 1)	`PTP-BOTH`	`PTP-BOTH,PTP-PTP,PTP-1PPS,SyncE-BOTH,SyncE-PTP,SyncE-1PPS`
`--PtpPoints`	Specifies which frequency points are to be tested with PTP input	`“0,1,2,3,4,5,6,7,8,9”`	`List` of points (see Note 2)
`--SyncEPoints`	Specifies which frequency points are to be tested with SyncE input	`“0,1,2,3,4,5,6,7,8,9,10,11”`	`List` of points (see Note 2)
`--ConfigFile`	Specifies a Tcl file containing configuration parameters (see Executing the Enhanced Noise Transfer Test)	“”	Valid Python file
`--SettlingTime`	Time for the device to lock to PTP supplied from the P-X Master (see Note 3).	`120`	Type: Integer Range: Positive Unit: Seconds
`--RecoveryTime`	Time for the device to recover after a change to the applied input noise (see Note 4).	`50`	Type: Integer Range: Positive Unit: Seconds
`--LimitsEnabled`	Specifies whether to check measured results against pre-defined limits. Disabling limit checking allows users to use their own limits.	`TRUE`	`TRUE, FALSE`

`--MasterMcastAddress`	Specifies whether the Forwardable Ethernet multicast address (01:1b:19:00:00:00) or non Forwardable address (01:80:c2:00:00:0e) should be used by the master	`FORWARDABLE`	`FORWARDABLE,`
			`NONFORWARDABLE`

`--SlaveMcastAddress`	Specifies whether the Forwardable Ethernet multicast address (01:1b:19:00:00:00) or non Forwardable address (01:80:c2:00:00:0e) should be used by the slave	`FORWARDABLE`	`FORWARDABLE,`
			`NONFORWARDABLE`

`--Graphing`	Enables/disables UI presentation of results	`FALSE`	`TRUE, FALSE`

Setting	Description	Default	Allowed Values
`--GraphingYAxis`	When **`Graphing`** is `TRUE`, plot results as either Gain(dB) or Amplitude(ns)	`GAIN`	`AMPLITUDE, GAIN`
`--Debug`	Enables/disables additional debugging output to the console	`FALSE`	`TRUE, FALSE`
`--ReProcess`	See "Re-Processing Previously Captured Data" below	`FALSE`	`TRUE, FALSE`
`--RemoteCaptureFolder`	See Using the Script Remotely	`C:/Desktop`	
`--RemoteProfileFolder`	See Using the Script Remotely	`C:/Desktop`	
`--Filter`	For Calnex use only. Do not change	`LSQ`	NA
`--SimulatedData`	For Calnex use only. Do not change	`FALSE`	NA

Table 6: Script Settings

Notes:

1. **`Type:`** Specifies the stimulus (PTP or SyncE) to be applied to the DUT input and the measurements to be performed on the DUT output. The measurements can be PTP, 1pps or BOTH.
2. **`PtpPoints / SyncEPoints:`** A subset of frequencies can be selected using a list of comma seperated delimited string of numbers indexed from 0. For example, to test only 0.0071825Hz and 0.4925 Hz, set `PtpPoints` to "`1,7`". To test all frequencies use the default setting from *Defaults.py*
3. **`SettlingTime:`** This should be long enough for the time error measurement to become stable, otherwise the results will be incorrect. This time will vary between DUTs.
4. **`RecoveryTime:`** A fixed period of time to allow the DUT to recover from each change of noise on the DUT input (i.e. for each frequency point). This time will very between DUTs. This recovery time will not be included in the measurement, but the measurement capture duration does include this time.

The defaults listed in the table above are contained in *Defaults.py*. This file may be modified by the user – the values in the table above are the defaults defined in the original file in the script install.

Paragon-X is automatically configured for the ITU-T G.8275.1 Phase Profile as part of the script initialization process; please ensure that the DUT is also configured for the G.8275.1 profile.

Executing the Enhanced Noise Transfer Test

Setting Configuration Parameters

The configuration parameters can be set in three different ways:

- Defaults: defined in *Defaults.py*
- User-specifed configuration file
- Command line arguments

The precedence order is the reverse of the above i.e. command line arguments over-ride settings in any configuration file which over-ride the defaults.

If no setting is specified in the command line arguments or configuration file, then the default value from *Defaults.py* will be used.

Note: The *Defaults.py* file may be edited by the user. However, for correct operation, ALL settings must be given a default value. Do not delete settings from this file.

Executing From the Command Line

Launch the Paragon-X application on the PC and run ***cmd.exe*** to launch a shell. Change the working directory (*cd*) to the script location. For example:

```
cd “C:\scripts\Script G.8273.2”
```

The script should now be run by the Python script (G.8273.2.py) with the required arguments as command line parameters. The arguments are passed to the script as a simple list of <argument value> pairs (separated by spaces). Some examples are shown below:

```
python G.8273.2.py --InstrIp “192.168.4.100” --LineRate “1GBE” --Interface “OPTICAL”
```

The command line arguments specify the instrument IP address, the line rate and the interface to be used. All other settings are defaults (specified in the *Defaults.py* file).

```
python G.8273.2.py --Type “PTP-1PPS” --Reference “EXT_10M” --ConfigFile “config.py”
```

The command line arguments specify the test type, the frequency reference. Other settings are defined in the *config.py* file. Any settings not specified on the command line or in the configuration file are defaults (specified in the *Defaults.py* file).

Executing From a Test Program

If the script is being integrated into another test program, then add the following code to the other program:

```
from subprocess import call

call(["python", "G.8273.2.py"])
```

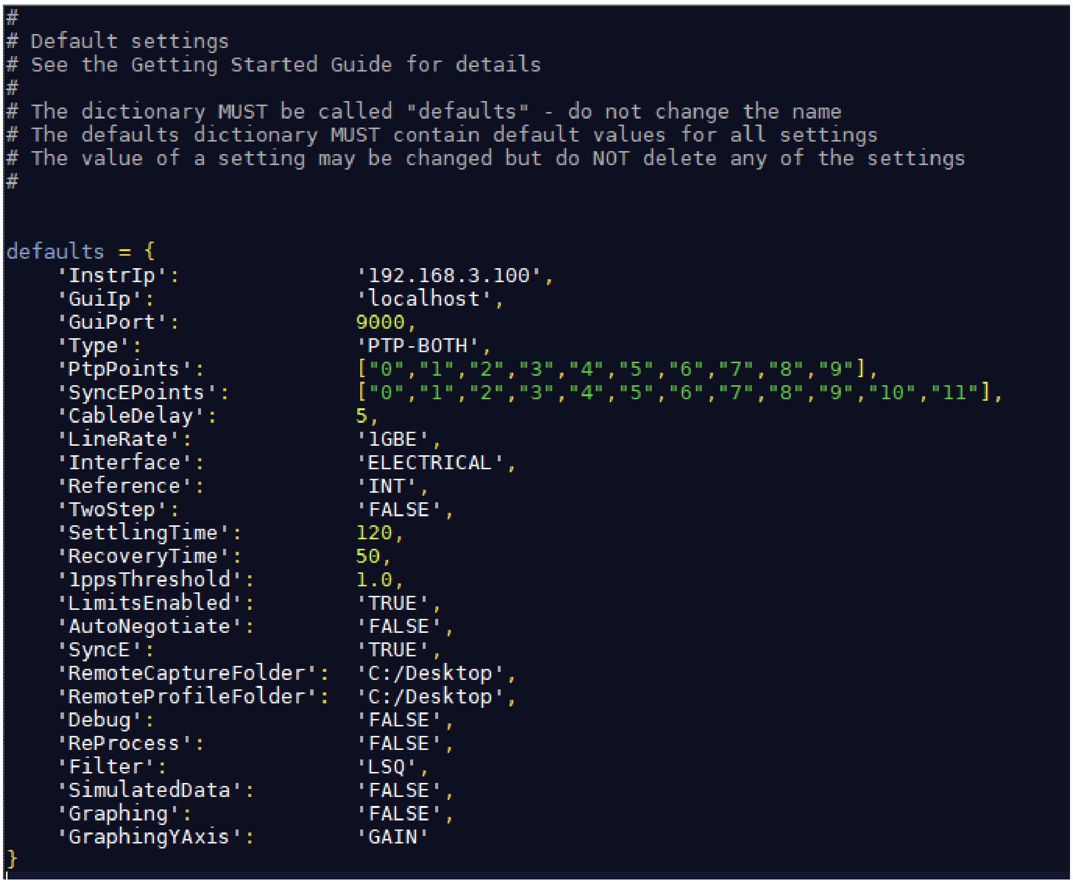
No arguments are required since all settings are defined in the *Defaults.py* file.

The Defaults.py File

The *Defaults.py* file defines the default settings used by the script. This file contains a Python dictionary defining the default settings used by the script. This file may be modified by the user if required.

The dictionary in this file **<u>must</u>** be called “`defaults`”. In addition, the file **<u>must</u>** define default values for all parameters.

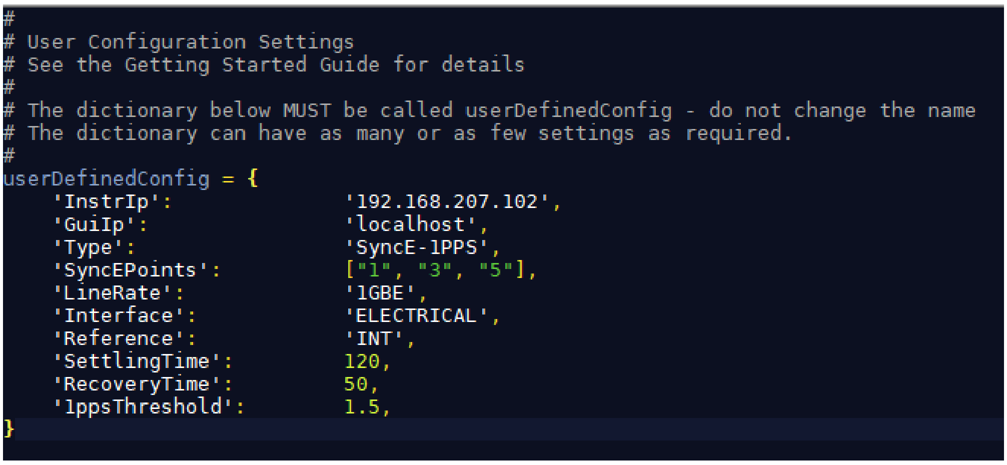
The *Defaults.py* file included in the install contains:



```
#
# Default settings
# See the Getting Started Guide for details
#
# The dictionary MUST be called "defaults" - do not change the name
# The defaults dictionary MUST contain default values for all settings
# The value of a setting may be changed but do NOT delete any of the settings
#


defaults = {
    'InstrIp':              '192.168.3.100',
    'GuiIp':                'localhost',
    'GuiPort':              9000,
    'Type':                 'PTP-BOTH',
    'PtpPoints':            ["0","1","2","3","4","5","6","7","8","9"],
    'SyncEPoints':          ["0","1","2","3","4","5","6","7","8","9","10","11"],
    'CableDelay':           5,
    'LineRate':             '1GBE',
    'Interface':            'ELECTRICAL',
    'Reference':            'INT',
    'TwoStep':              'FALSE',
    'SettlingTime':         120,
    'RecoveryTime':         50,
    '1ppsThreshold':        1.0,
    'LimitsEnabled':        'TRUE',
    'AutoNegotiate':        'FALSE',
    'SyncE':                'TRUE',
    'RemoteCaptureFolder':  'C:/Desktop',
    'RemoteProfileFolder':  'C:/Desktop',
    'Debug':                'FALSE',
    'ReProcess':            'FALSE',
    'Filter':               'LSQ',
    'SimulatedData':        'FALSE',
    'Graphing':             'FALSE',
    'GraphingYAxis':        'GAIN'
}
```

Configuration Files

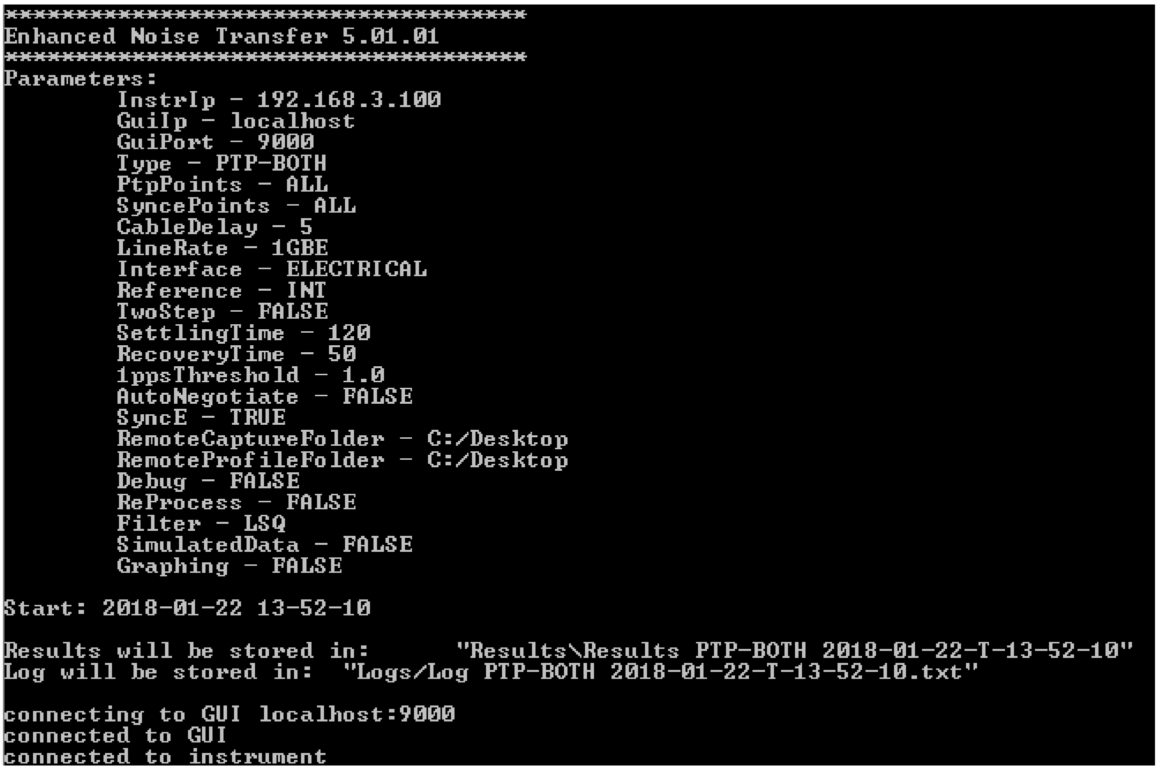
A configuration file can also be used to set parameters. This file <u>must</u> contain a Python dictionary called “`userDefinedConfig`”. This file can define as many or as few parameters as required. An example configuration file (*ExampleConfig.py*) is included in the install. This contains:



```
#
# User Configuration Settings
# See the Getting Started Guide for details
#
# The dictionary below MUST be called userDefinedConfig - do not change the name
# The dictionary can have as many or as few settings as required.
#
userDefinedConfig = {
    'InstrIp':                '192.168.207.102',
    'GuiIp':                  'localhost',
    'Type':                   'SyncE-1PPS',
    'SyncEPoints':            ["1", "3", "5"],
    'LineRate':               '1GBE',
    'Interface':              'ELECTRICAL',
    'Reference':              'INT',
    'SettlingTime':           120,
    'RecoveryTime':           50,
    '1ppsThreshold':          1.5,
}
```

Console Output

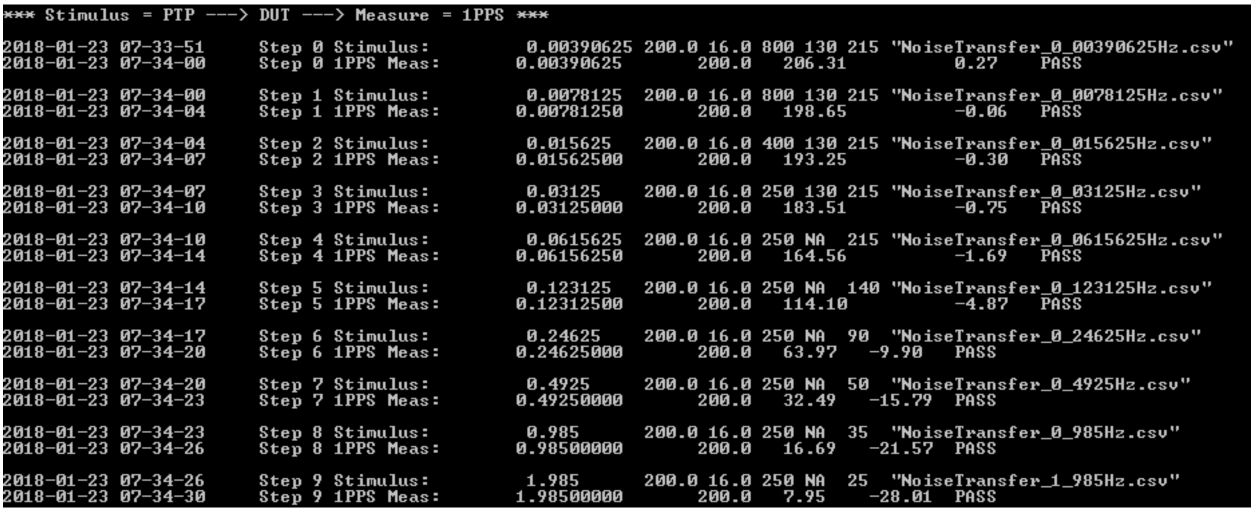
The initial console output from a script execution is shown below:



```
*************************************
Enhanced Noise Transfer 5.01.01
*************************************
Parameters:
        InstrIp - 192.168.3.100
        GuiIp - localhost
        GuiPort - 9000
        Type - PTP-BOTH
        PtpPoints - ALL
        SyncePoints - ALL
        CableDelay - 5
        LineRate - 1GBE
        Interface - ELECTRICAL
        Reference - INT
        TwoStep - FALSE
        SettlingTime - 120
        RecoveryTime - 50
        1ppsThreshold - 1.0
        AutoNegotiate - FALSE
        SyncE - TRUE
        RemoteCaptureFolder - C:/Desktop
        RemoteProfileFolder - C:/Desktop
        Debug - FALSE
        ReProcess - FALSE
        Filter - LSQ
        SimulatedData - FALSE
        Graphing - FALSE

Start: 2018-01-22 13-52-10

Results will be stored in:      "Results\Results PTP-BOTH 2018-01-22-T-13-52-10"
Log will be stored in:  "Logs/Log PTP-BOTH 2018-01-22-T-13-52-10.txt"

connecting to GUI localhost:9000
connected to GUI
connected to instrument
```

The settings being used by the script will be printed to the console before the test begins. All settings are printed regardless of where they were defined (command line, config file, defaults).

As the test progresses, the results will also be written to the console.

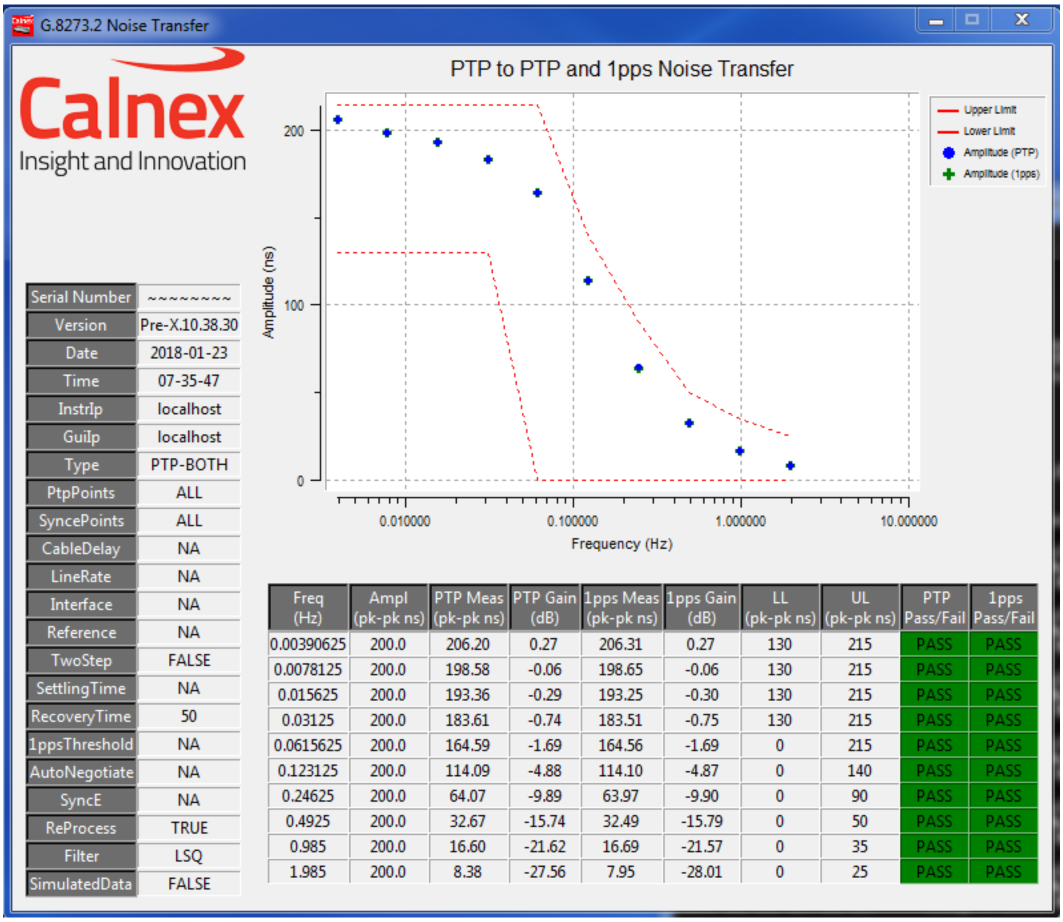


```
*** Stimulus = PTP ---> DUT ---> Measure = 1PPS ***

2018-01-23 07-33-51     Step 0 Stimulus:         0.00390625 200.0 16.0 800 130 215 "NoiseTransfer_0_00390625Hz.csv"
2018-01-23 07-34-00     Step 0 1PPS Meas:       0.00390625       200.0   206.31          0.27    PASS

2018-01-23 07-34-00     Step 1 Stimulus:         0.0078125  200.0 16.0 800 130 215 "NoiseTransfer_0_0078125Hz.csv"
2018-01-23 07-34-04     Step 1 1PPS Meas:       0.00781250       200.0   198.65          -0.06   PASS

2018-01-23 07-34-04     Step 2 Stimulus:         0.015625   200.0 16.0 400 130 215 "NoiseTransfer_0_015625Hz.csv"
2018-01-23 07-34-07     Step 2 1PPS Meas:       0.01562500       200.0   193.25          -0.30   PASS

2018-01-23 07-34-07     Step 3 Stimulus:         0.03125    200.0 16.0 250 130 215 "NoiseTransfer_0_03125Hz.csv"
2018-01-23 07-34-10     Step 3 1PPS Meas:       0.03125000       200.0   183.51          -0.75   PASS

2018-01-23 07-34-10     Step 4 Stimulus:         0.0615625  200.0 16.0 250 NA  215 "NoiseTransfer_0_0615625Hz.csv"
2018-01-23 07-34-14     Step 4 1PPS Meas:       0.06156250       200.0   164.56          -1.69   PASS

2018-01-23 07-34-14     Step 5 Stimulus:         0.123125   200.0 16.0 250 NA  140 "NoiseTransfer_0_123125Hz.csv"
2018-01-23 07-34-17     Step 5 1PPS Meas:       0.12312500       200.0   114.10          -4.87   PASS

2018-01-23 07-34-17     Step 6 Stimulus:         0.24625    200.0 16.0 250 NA  90  "NoiseTransfer_0_24625Hz.csv"
2018-01-23 07-34-20     Step 6 1PPS Meas:       0.24625000       200.0   63.97   -9.90   PASS

2018-01-23 07-34-20     Step 7 Stimulus:         0.4925     200.0 16.0 250 NA  50  "NoiseTransfer_0_4925Hz.csv"
2018-01-23 07-34-23     Step 7 1PPS Meas:       0.49250000       200.0   32.49   -15.79  PASS

2018-01-23 07-34-23     Step 8 Stimulus:         0.985      200.0 16.0 250 NA  35  "NoiseTransfer_0_985Hz.csv"
2018-01-23 07-34-26     Step 8 1PPS Meas:       0.98500000       200.0   16.69   -21.57  PASS

2018-01-23 07-34-26     Step 9 Stimulus:         1.985      200.0 16.0 250 NA  25  "NoiseTransfer_1_985Hz.csv"
2018-01-23 07-34-30     Step 9 1PPS Meas:       1.98500000       200.0   7.95    -28.01  PASS
```

UI

Using the Calnex defaults, the graphing option is disabled (`Graphing` is set to `FALSE` in *Defaults.tcl/ Defaults.py*) file.

When enabled, the UI allows the progress of the test to be monitored as well as providing a bode plot in which the results are presented.

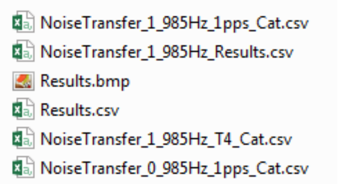


Serial Number	~~~~~~~~
Version	Pre-X.10.38.30
Date	2018-01-23
Time	07-35-47
InstrIp	localhost
GuiIp	localhost
Type	PTP-BOTH
PtpPoints	ALL
SyncePoints	ALL
CableDelay	NA
LineRate	NA
Interface	NA
Reference	NA
TwoStep	FALSE
SettlingTime	NA
RecoveryTime	50
1ppsThreshold	NA
AutoNegotiate	NA
SyncE	NA
ReProcess	TRUE
Filter	LSQ
SimulatedData	FALSE

Freq (Hz)	Ampl (pk-pk ns)	PTP Meas (pk-pk ns)	PTP Gain (dB)	1pps Meas (pk-pk ns)	1pps Gain (dB)	LL (pk-pk ns)	UL (pk-pk ns)	PTP Pass/Fail	1pps Pass/Fail
0.00390625	200.0	206.20	0.27	206.31	0.27	130	215	PASS	PASS
0.0078125	200.0	198.58	-0.06	198.65	-0.06	130	215	PASS	PASS
0.015625	200.0	193.36	-0.29	193.25	-0.30	130	215	PASS	PASS
0.03125	200.0	183.61	-0.74	183.51	-0.75	130	215	PASS	PASS
0.0615625	200.0	164.59	-1.69	164.56	-1.69	0	215	PASS	PASS
0.123125	200.0	114.09	-4.88	114.10	-4.87	0	140	PASS	PASS
0.24625	200.0	64.07	-9.89	63.97	-9.90	0	90	PASS	PASS
0.4925	200.0	32.67	-15.74	32.49	-15.79	0	50	PASS	PASS
0.985	200.0	16.60	-21.62	16.69	-21.57	0	35	PASS	PASS
1.985	200.0	8.38	-27.56	7.95	-28.01	0	25	PASS	PASS

The `GraphingYAxis` argument can be used to set the Y axis to either `GAIN` or `AMPLITUDE`.

When the UI is enabled, a screenshot (.bmp) of the UI is taken when the test completes. This can be found in the *Results* folder:


NoiseTransfer_1_985Hz_1pps_Cat.csv
NoiseTransfer_1_985Hz_Results.csv
Results.bmp
Results.csv
NoiseTransfer_1_985Hz_T4_Cat.csv
NoiseTransfer_0_985Hz_1pps_Cat.csv

Test Execution

Test execution follows the following steps:

- Reset the Paragon-X
- Configure the Paragon-X
- Start MSE
- Wait for `SettlingTime` seconds to ensure that the DUT has locked to the input PTP.

- Measure the DUT noise generation with no noise applied to the input. This is useful for later analysis of the results. The baseline noise generation measurement takes 120 seconds.
- For each step in the PTP or SyncE table
    - Apply PTP or SyncE noise to the DUT input from the Paragon-X Master
    - Capture the PTP and/or 1pps output from the DUT for a fixed duration
    - Remove `RecoveryTime` seconds from the capture. This makes sure that the DUTs response to the change in noise does not affect the results.
    - Process the data using Least Squares estimation
    - Calculate the gain
    - Present the results
- Stop MSE

The script may take a significant time to complete. For example, a complete `SyncE-1PPS` test run will take slightly longer than 4370 seconds (70+ minutes). See Table 1 and Table 2 for the duration of each frequency step.

Progress can be tracked by observing the results of each individual test point as they are processed and displayed in the Tcl console or on the UI.

For `PTP-PTP`, `PTP-1PPS` and `PTP-BOTH` noise transfer, the ENT script applies Time Error noise to *Sync* messages at the defined frequencies with a peak-to-peak amplitude of double that defined in Table 1 (e.g. for the default configuration, the amplitude is 400ns so as to acheive a 2-way Time Error noise of 200ns) using a message rate of 16pkt/s.

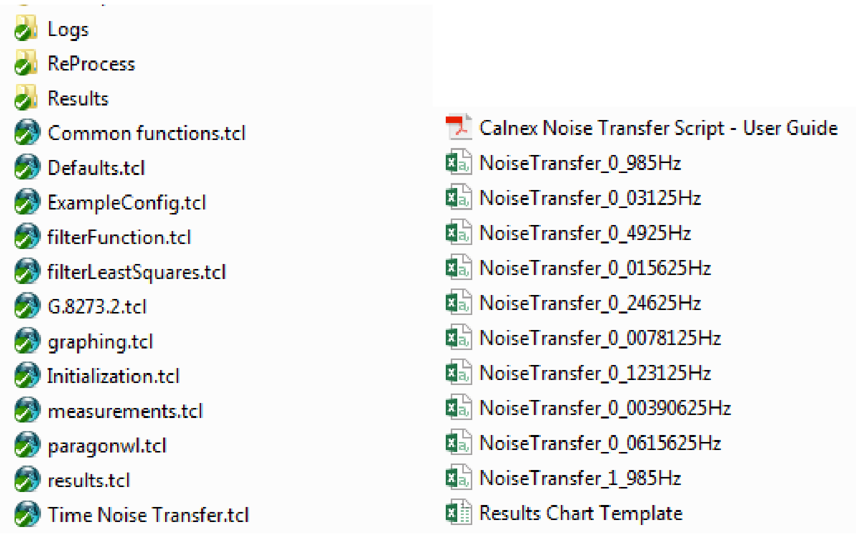
For `SyncE-PTP`, `SyncE-1PPS` and `SyncE-BOTH` noise transfer, the ENT script applies sinusoidal wander patterns to the SyncE output from the Master at the frequencies and peak-to-peak amplitudes as defined in Table 2.

File Locations and Results

The script will generate new results files and will copy capture files each time the script is run. These files are not automatically deleted.

All files are stored relative to the folder in which the script is being run.

The root folder contains the Tcl script files, the PTP profiles, the Results Chart Template and the ENT user guide (this document):

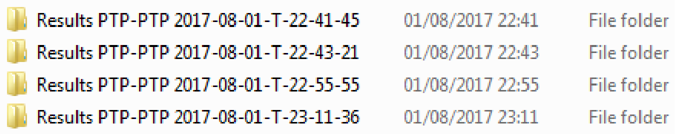


```
Logs
ReProcess
Results
Common functions.tcl
Defaults.tcl
ExampleConfig.tcl
filterFunction.tcl
filterLeastSquares.tcl
G.8273.2.tcl
graphing.tcl
Initialization.tcl
measurements.tcl
paragonwl.tcl
results.tcl
Time Noise Transfer.tcl
Calnex Noise Transfer Script - User Guide
NoiseTransfer_0_985Hz
NoiseTransfer_0_03125Hz
NoiseTransfer_0_4925Hz
NoiseTransfer_0_015625Hz
NoiseTransfer_0_24625Hz
NoiseTransfer_0_0078125Hz
NoiseTransfer_0_123125Hz
NoiseTransfer_0_00390625Hz
NoiseTransfer_0_0615625Hz
NoiseTransfer_1_985Hz
Results Chart Template
```

All results files and capture files will be stored in the ***Results*** folder in the same directory as the script files. Each execution of the script will create a sub-folder using the following naming convention:

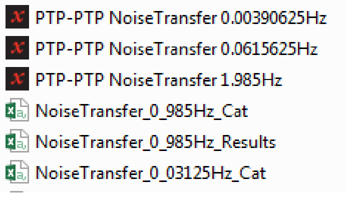
```
Results <Type> <YYYY-MM-DD-T-HH-MM-SS>
```

Where `<Type>` is the test type e.g. `PTP-BOTH`. For example:



```
Results PTP-PTP 2017-08-01-T-22-41-45    01/08/2017 22:41    File folder
Results PTP-PTP 2017-08-01-T-22-43-21    01/08/2017 22:43    File folder
Results PTP-PTP 2017-08-01-T-22-55-55    01/08/2017 22:55    File folder
Results PTP-PTP 2017-08-01-T-23-11-36    01/08/2017 23:11    File folder
```

Each sub-folder will contain the capture files and the generated (interim) results files (in csv format). For example:



```
PTP-PTP NoiseTransfer 0.00390625Hz
PTP-PTP NoiseTransfer 0.0615625Hz
PTP-PTP NoiseTransfer 1.985Hz
NoiseTransfer_0_985Hz_Cat
NoiseTransfer_0_985Hz_Results
NoiseTransfer_0_03125Hz_Cat
```

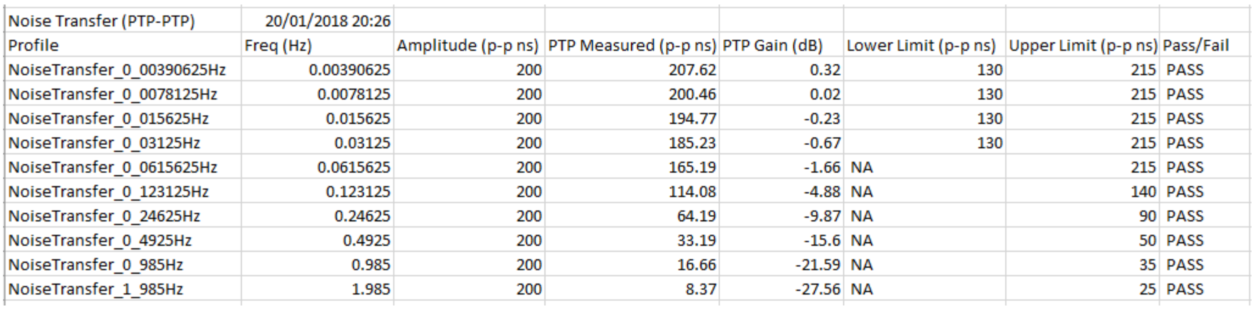
This folder will also contain the final results file with the overall results for all test points. This file is named ***Results.csv***. If the UI was enabled (`Graphing` set to `TRUE`), then the UI screenshot will also be stored in the ***Results*** folder. This is named ***Results.bmp***.

For each profile used in the script, the following information will be recorded in the csv results file:

- Profile filename (for `PTP-PTP`, `PTP-1PPS` and `PTP-BOTH`) or frequency (`SyncE-PTP`, `SyncE-1PPS` and `SyncE-BOTH`)
- Frequency (Hz)

- Input Amplitude (pk-pk ns)
- Measured Amplitude (pk-pk ns): The noise transferred to the output of the DUT
- Gain (dB): The ratio of measured amplitude (output from the DUT) to input amplitude (generated by Paragon-X).
- Lower Limit (pk-pk ns): The lower limit from G.8273.2 Amendment 1.
- Upper Limit (pk-pk ns): The upper limit from G.8273.2 Amendment 1.
- Pass / Fail: Will be one of Pass, Fail (lower), Fail (upper)

For example, a `PTP-PTP` results file contains the following information:



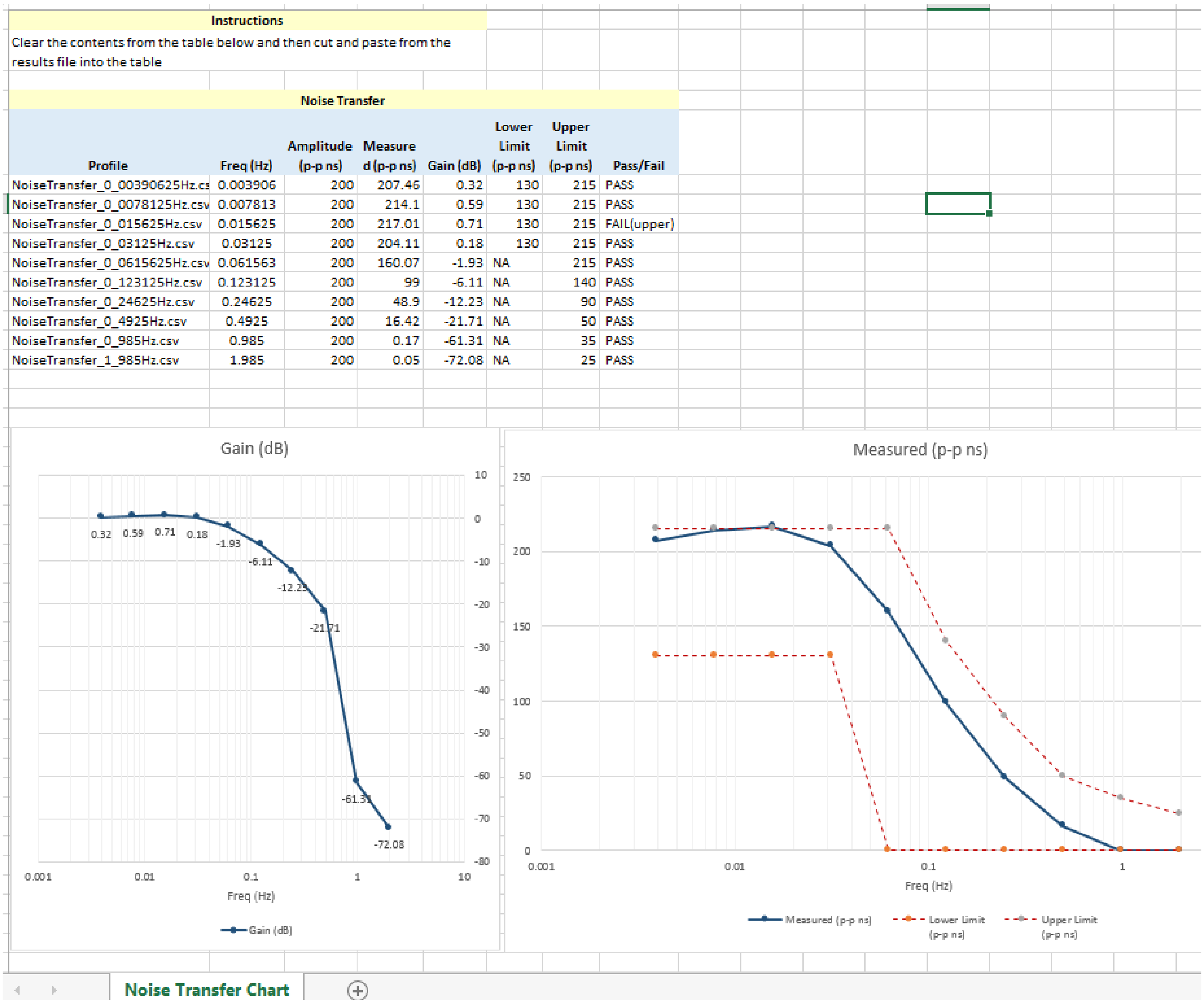
Noise Transfer (PTP-PTP)	20/01/2018 20:26						
Profile	Freq (Hz)	Amplitude (p-p ns)	PTP Measured (p-p ns)	PTP Gain (dB)	Lower Limit (p-p ns)	Upper Limit (p-p ns)	Pass/Fail
NoiseTransfer_0_00390625Hz	0.00390625	200	207.62	0.32	130	215	PASS
NoiseTransfer_0_0078125Hz	0.0078125	200	200.46	0.02	130	215	PASS
NoiseTransfer_0_015625Hz	0.015625	200	194.77	-0.23	130	215	PASS
NoiseTransfer_0_03125Hz	0.03125	200	185.23	-0.67	130	215	PASS
NoiseTransfer_0_0615625Hz	0.0615625	200	165.19	-1.66	NA	215	PASS
NoiseTransfer_0_123125Hz	0.123125	200	114.08	-4.88	NA	140	PASS
NoiseTransfer_0_24625Hz	0.24625	200	64.19	-9.87	NA	90	PASS
NoiseTransfer_0_4925Hz	0.4925	200	33.19	-15.6	NA	50	PASS
NoiseTransfer_0_985Hz	0.985	200	16.66	-21.59	NA	35	PASS
NoiseTransfer_1_985Hz	1.985	200	8.37	-27.56	NA	25	PASS

If `PTP-BOTH` or `SyncE-BOTH` is selected, the the results for both PTP and 1pps will be included in the results.

Presenting the Results

If the UI has been enabled, then a screenshot of the results are saved as Results.bmp in the Results folder. This contains the detailed results in a table as well as a bode plot.

In addition, Excel can be used to plot the transfer characteristic of the DUT. An excel spreadsheet (***Results Chart Template.xlsx***) is included in the test bundle to assist in creating a bode plot of the results. Data from a ***Results.csv*** file can be copied and pasted into the template to generate the plots shown below:



Instructions

Clear the contents from the table below and then cut and paste from the results file into the table

Noise Transfer

Profile	Freq (Hz)	Amplitude (p-p ns)	Measured (p-p ns)	Gain (dB)	Lower Limit (p-p ns)	Upper Limit (p-p ns)	Pass/Fail
NoiseTransfer_0_00390625Hz.cs	0.003906	200	207.46	0.32	130	215	PASS
NoiseTransfer_0_0078125Hz.csv	0.007813	200	214.1	0.59	130	215	PASS
NoiseTransfer_0_015625Hz.csv	0.015625	200	217.01	0.71	130	215	FAIL(upper)
NoiseTransfer_0_03125Hz.csv	0.03125	200	204.11	0.18	130	215	PASS
NoiseTransfer_0_0615625Hz.csv	0.061563	200	160.07	-1.93	NA	215	PASS
NoiseTransfer_0_123125Hz.csv	0.123125	200	99	-6.11	NA	140	PASS
NoiseTransfer_0_24625Hz.csv	0.24625	200	48.9	-12.23	NA	90	PASS
NoiseTransfer_0_4925Hz.csv	0.4925	200	16.42	-21.71	NA	50	PASS
NoiseTransfer_0_985Hz.csv	0.985	200	0.17	-61.31	NA	35	PASS
NoiseTransfer_1_985Hz.csv	1.985	200	0.05	-72.08	NA	25	PASS

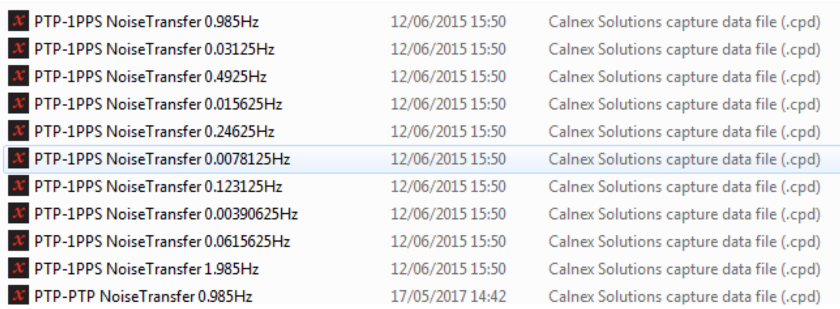
Note: The template supports only a single measurement. If both PTP and 1pps have been tested, plots must be generated individually.

Re-Processing Previously Captured Data

Using `ReProcess TRUE` allows previously captured data to be re-analysed. The capture files to be re-analysed must be stored in a folder called ***ReProcess*** in the same directory as the script. The files should be named:

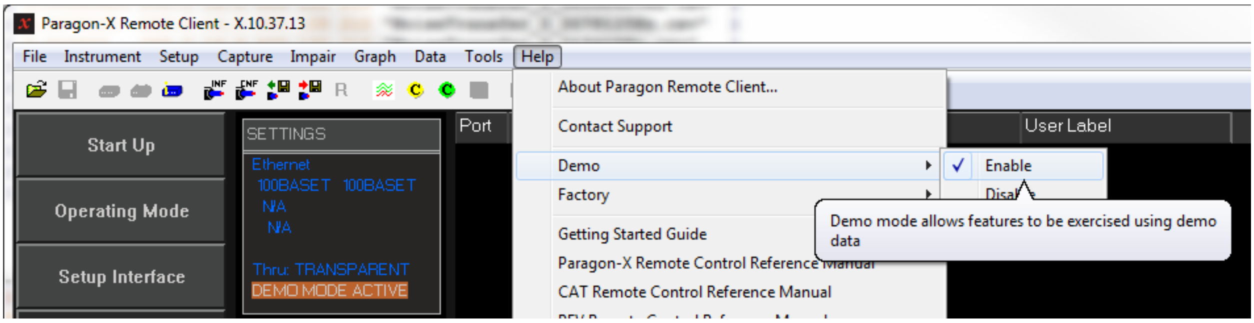
`<Type>` NoiseTransfer `<freq>`Hz.cpd

For example:



PTP-1PPS NoiseTransfer 0.985Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-1PPS NoiseTransfer 0.03125Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-1PPS NoiseTransfer 0.4925Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-1PPS NoiseTransfer 0.015625Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-1PPS NoiseTransfer 0.24625Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-1PPS NoiseTransfer 0.0078125Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-1PPS NoiseTransfer 0.123125Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-1PPS NoiseTransfer 0.00390625Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-1PPS NoiseTransfer 0.0615625Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-1PPS NoiseTransfer 1.985Hz	12/06/2015 15:50	Calnex Solutions capture data file (.cpd)
PTP-PTP NoiseTransfer 0.985Hz	17/05/2017 14:42	Calnex Solutions capture data file (.cpd)

Before running the script, the Paragon-X application must be in Demo Mode:



Demo mode allows features to be exercised using demo data

The script can then be run in a similar way to the actual test run e.g.

TCL:

```
tclsh86 G.8273.2.tcl Type PTP-1PPS ReProcess TRUE RecoveryTime 20
```

Python:

```
python G.8273.2.py --Type “PTP-1PPS” --ReProcess “TRUE”  --RecoveryTime 20
```

The results that are produced when re-processing are the same as those for a normal test run.

Note: The recovery time used for re-processing can be different from that used for the original test run. This may affect the results.

3 Troubleshooting

For support using ENT, please contact your local Calnex Sales Representative or email support@calnex.com.

Before requestiung support, please make sure that you can successfully make manual Time Error measurements on your DUT with the same configuration as is used by the script e.g. using the G.8275.1 profile in conjunction with the script settings you use. If you are unable to make Time Error measurements manually, then the script will fail.

When requesting support, please provide the following:

- The version numbers of the Paragon-X and CAT applications
- **A zip file containing the complete contents of your script folder and all sub-folders**. Note that, if the ENT is not running on the same machine as Paragon-X, the files from both the local and remote hosts will need to be combined.
- A screenshot or file containg the script output generated during the test run. If possible, this should be generated using `Debug TRUE`.

If you are having issues running the script, please review the list below to establish whether these may be the cause of your problem:

- The script stops unexpectedly:

  Please make sure that the PC running the script or hosting the Paragon-X application is set such that is will not automatically hibernate or sleep. The scripts run for a significant period of time and this is generally longer than the default period for automatic hibernation.

- The script cannot connect to the Paragon-X application running on a remote host

  Please make sure that the *local host* has access to the *remote host*. This can be confirmed by pinging the IP address of the *remote host* from the *local host*.

- The script cannot retrieve the results from the *remote host*

  Results are read from the *remote host* using port 1337. Please ensure that this port must be open and accessible from the *local host*.

4 Using the Script Remotely

Installation

The contents of the zip file should be extracted to a folder of your choice on any machine.

Some of the extracted files must be located on the *local host* and some on the remote Windows PC. On the *remote host*, the PTP profiles must be stored in a folder accessible to the Paragon-X application.The script files must be saved to a folder on the *local host*. The files concerned are shown in the figure below.

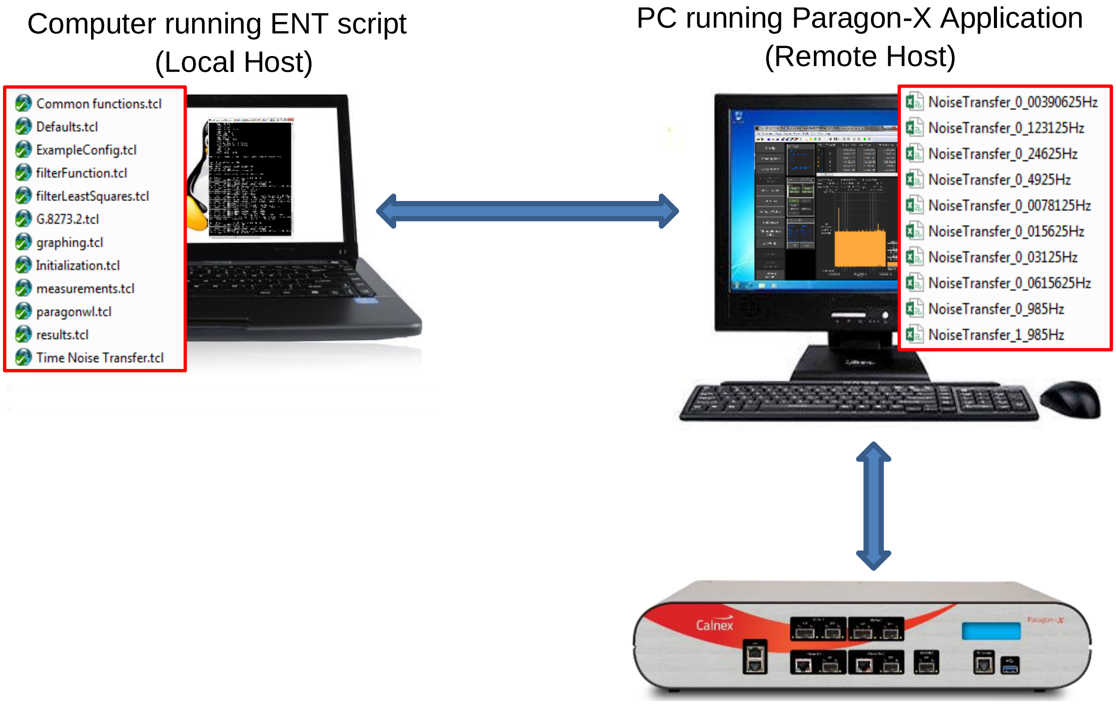


Figure 5: Location of Files for Remote Use

Configuration

In addition to the settings described in Configuration above, there are three settings that are particularly relevant when running remotely:

Setting	Description	Default	Allowed Values
GuiIp	The IP address of the PC hosting the Paragon-X application	locahost	Valid IP address (see note 0)
RemoteCaptureFolder	The folder in which to store capture files on the remote Windows PC (***Remote Host***).	C:/Desktop	See note 2
RemoteProfileFolder	The folder which contains the profiles on the remote Windows PC (***Remote Host***).	C:/Desktop	See note 3.

Table 7: Settings for Remote Use

1. **`GuiIp`**: To run remotely, this must be the IP address of the remote host running the Paragon-X application.
2. **`RemoteCaptureFolder`**: This is the folder on the *remote host* where capture files will be stored. This folder must exist prior to running the script. The script cannot create this folder since it is on the *remote host*. The folder must be able to be written by the Paragon-X application.
3. **`RemoteProfileFolder`**: This is the folder on the *remote host* where the profiles have been stored (see Installation above). This folder must be able to be read by the Paragon-X application.

Executing the Enhanced Noise Transfer Test using a Remote Host

Execution of the script for remote use is essentially the same as for local use but the three configuration parameters above must be set correctly. The example below explains how this should be done using the command line only. It is also possible to do this by modifying the *Defaults.tcl/ Defaults.py* file or using a configuration file (see Executing the Enhanced Noise Transfer Test above for details).

The Paragon-X application should be running on the ***remote host*.**

Use ***cmd.exe*** to open a shell on the ***local host*** and change the working directory (*cd*) to the script location. For example:

```
cd “C:\scripts\Script G.8273.2”
```

TCL:

The script should now be run by launching a Tcl shell passing the script name with the required arguments as command line parameters.

The arguments are passed to the script as a simple list of <argument value> pairs (separated by spaces). For remote use, `GuiIp` must be set to the IP address of the ***local host*** running the Paragon-X application.

Some examples are shown below (note that there are extra line breaks for formatting only

```
tclsh
G.8273.2.tcl InstrIp 192.168.4.100 GuiIp 192.168.4.105 LineRate 1GBE Interface OPTICAL
RemoteCaptureFolder “C:/Desktop/ENTScriptResults” RemoteProfileFolder “C:/Desktop/ENTProfiles”

tclsh G.8273.2.tcl Type PTP-1PPS GuiIp 192.168.4.105 Reference EXT_10M RemoteCaptureFolder
“C:/Desktop/ENTScriptResults”
```

Python:

The script can be launched directly from the command line using the associated argurments as command line parameters

The arguments are passed to the script as a simple list of <argument value> pairs (separated by spaces). For remote use, `GuiIp` must be set to the IP address of the ***local host*** running the Paragon-X application.

Some examples are shown below (note that there are extra line breaks for formatting only

```
python G.8273.2.py --InstrIp “192.168.4.100” --GuiIp “192.168.4.105” --LineRate “1GBE” --
Interface “OPTICAL” --RemoteCaptureFolder “C:/Desktop/ENTScriptResults” --RemoteProfileFolder
“C:/Desktop/ENTProfiles”
```

```
python G.8273.2.py --Type “PTP-1PPS” --GuiIp “192.168.4.105” --Reference “EXT_10M”
--RemoteCaptureFolder “C:/Desktop/ENTScriptResults”
```

The test will then be executed as described in

Test Execution above.

File Locations and Results when using a Remote Host

All Tcl files are stored on the ***local host***. All profile files are stored on the ***remote host*** (in the folder specified by the `RemoteProfileFolder` argument).

All result (csv) files will be stored in the ***Results*** folder in the same directory as the script files on the *local host*. Each execution of the script will create a sub-folder using the same naming convention as described in File Locations and Results.

The structure and content of ***Results.csv*** will be the same as when running locally.

Capture files are stored on the ***remote host*** in the folder specified by the `RemoteCaptureFolder` parameter.

Calnex Solutions Ltd
Oracle Campus
Linlithgow
EH49 7LR
United Kingdom

t: +44 (0) 1506 671 416
e: info@calnexsol.com

www.calnexsol.com

Document CX5036 v7.4 February 21