APPLICATION NOTE

Using the WaveAnalyzer Analysis Server

1. Scope and Overview

The WaveAnalyzer Analysis Server application packaged with WaveAnalyzer software version 1.7 and onwards provides additional programming functionality for the WaveAnalyzer 1500S, including access to the OSNR, WDM and Wavelength Meter functions provided in the WaveAnalyzer GUI.

The WaveAnalyzer Analysis Server is a standalone executable providing a RESTful style Web service API to provide access to measurement data and

This Application Note provides details on how to connect to, and program using, the Analysis Server. Details of each command are explained and programming examples provided.

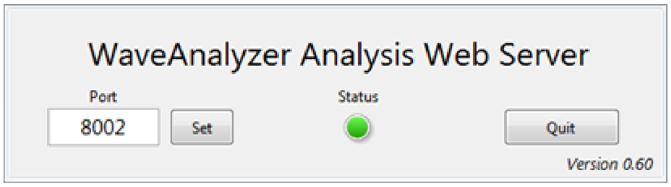
It is recommended that you first read the Application Note entitles “Controlling the WaveShaper 1500S using the Web API” before using the techniques described in this Application Note.

2. Starting and Configuring the Server

The server application can be started via the start menu: Start -> Finisar -> WaveAnalyzer Analysis Server

Alternatively, the server can be started programmatically by running the server executable within an automated test environment, from the following default installation location:
C:\Program Files (x86)\Finisar\WaveAnalyzer\AnalysisServer\WA-AnalysisServer.exe

The server application has the following user interface:



WaveAnalyzer Analysis Web Server

Port: 8002

Figure 2-1 WaveAnalyzer Analysis Server GUI

Configure the server port by entering a desired port number into the *Port* input and clicking the *Set* button. Changing the port number requires a restart of the application.

The *Status* indicator indicates whether the Web server is up and running successfully (green), or not (red). Note that the status indicator is red on start-up of the server application, and remains red for up to 10 seconds, while the Web server loads.

3. Analysis Server command format

The HTTP command format for the WaveAnalyzer Analysis Server is as follows:

http://<server address>/analysis/<method>?<input parameters>

<server address> is the IP address of the WaveAnalyzer Analysis Server. Most often, this is *localhost:8002*, i.e. the server application runs on the local PC and is configured at port 8002.

<method> specifies which API method to perform, e.g. **data** for taking scans, or **osnr** for taking OSNR measurements. A question mark character (?) is used to separate the method and the input parameters.

<input parameters> specify a list of input parameters dictating how each command will operate. Some of these parameters are required and some are optional. The input parameter list is specified in the following format:

<parameter1>=<value1>&<parameter2>=<value2>&...

Each parameter is separated by an ampersand (&) character.

The following example demonstrates a typical HTTP command for a WaveAnalyzer 1500S:

http://localhost:8002/analysis/data?ip=192.168.0.3&averages=32&rbw=3000

This HTTP command contains the following details:

- The analysis server address is localhost, with port number 8002.
- The API method is **data**.
- The input parameters specify a device with an IP address of 192.168.0.3, 32 averages and a resolution bandwidth of 3000 MHz.

4. Input Parameters and Result Codes

The following input parameters are available for all API methods.

Input Parameter	Type	Default	Description
ip	Optional	169.254.3.8	The IP address of the WaveAnalyzer 1500S to be accessed.
averages	Optional	1	The number of measurements to scan and average before responding with measurement data or performing analysis functions.
scantype	Optional	measure	Determines what measurement data to use when performing analysis. The available options are: `measure` – The method issues a fresh measurement and performs the analysis using the resulting measurement data. The measurement data is saved into memory where it can be further interrogated using the `query` scan type. `query` – The method performs the analysis using measurement data saved in memory from the most recent measurement.

The following result codes are possible

Result Code	Description
0	Success
-38	Device not found
-60	Invalid input parameter
503	The server is unable to handle the request due to temporary overloading

5. Analysis Server API methods

5.1 Perform Scan and Download measurement data

Description: Commands the instrument to take a scan and downloads the measurement data once complete. An averaged scan is performed if the number of averages specified is larger than 1. A digital convolution is applied to the data if the specified resolution bandwidth is larger than the default value for the WaveAnalyzer 1500S of 150 MHz.

API Method: data

URL: http://<server address>/analysis/data?<input parameters>

Input Parameters:

Input Parameter	Type	Default	Unit	Description
rbw	Optional	150	MHz	The resolution bandwidth of the measurement.
shape	Optional	flattop	-	If a resolution bandwidth is specified larger than 150 MHz, then the measurement is digitally filtered. This parameter determines what filter method to use (these methods are equivalent to the options available in the WaveAnalyzer software): `flattop` – The filter is rectangular in shape. `gaussian` – The filter is Gaussian in shape.

Response: Tab-delimited text response with header. In case an error is encountered during execution, the server responds with a plain text error message containing an error code and error description.

Example:

```
http://localhost:8002/analysis/data?ip=169.254.3.8&aver
ages=32&rbw=3000
```

```
Model Type: WaveAnalyzer 1500S
Model Number: 1500S
Serial Number: WA000123
Firmware Version: 1.02
Software Version: N.A.
Scan ID: 13900205
Resolution Bandwidth: 3000 MHz
Creation Time: 2017-05-11T11:59:00

Data:
Frequency [MHz] Absolute Power [mdBm]   Power X-
Polarization [mdBm]  Power Y-Polarization [mdBm]
191100170       -34611  -36891  -37197
191100270       -34522  -36802  -37109
191100370       -34438  -36716  -37029
...
196377170       -46524  -49354  -48781
196377270       -46516  -49350  -48772
196377370       -46521  -49349  -48785
```

5.2 OSNR Measurements

Description: Take a 3-point or 6-point OSNR (Optical Signal-to-Noise Ratio) measurement.

API method: osnr

URL: http://<server address>/analysis/osnr?<input parameters>

Input Parameters:

Input Parameter	Type	Default	Unit	Description
frequencies	Required	-	MHz	A comma-separated list of frequencies denoting the frequency points used in the OSNR calculation. When three or six frequencies are provided, the algorithm uses the 3-point or 6-point OSNR algorithm respectively.

Response: JSON formatted response including the following fields:

JSON field	Unit	Description
osnr	mdB	The measured OSNR.
rc	-	Result code indicating whether the measurement was successful. See Section 4 for description of result codes.
msg	-	A message containing information about the request, e.g. a description of any errors.
valid	-	This flag is `true` when the OSNR measurement was valid. This flag returns `false` in case the power in the signal after noise subtraction becomes negative.

Example:

```
http://localhost:8002/analysis/osnr?ip=192.168.2.8&freq
uencies=192995000,193000000,193005000
```

```
{"rc":0,"msg":"Success","osnr":28206,"valid":true}
```

5.3 Wavelength Meter Measurements

Description: Take a wavelength meter measurement to determine center frequencies and integrated power values of multiple signals present within a trace.

API method: wm

URL: http://<server address>/analysis/wm?<input parameters>

Input Parameters:

Input Parameter	Type	Default	Unit	Description
threshold	Optional	-40000	mdBm	The threshold value used in the peak finding algorithm.
excursion	Optional	1000	mdB	The excursion value used in the peak finding algorithm.
mode	Optional	moments	-	The method used to calculate the center frequency. Available options include: `moments` – The center frequency is given as the spectral centroid, or 'center of mass' of the spectrum within the power integration window. `peak` – The center frequency is given by finding the point with the highest power value within the power integration window.
width	Optional	10000	MHz	The spectral width of the window in which power is integrated over. This window is centered on the peak frequency.
startfreq	Optional	Scan Start	MHz	The minimum frequency to be used by the peak finding algorithm. This is used in conjunction with stopfreq in order to specify a frequency range in which the peak finding algorithm operates over.
stopfreq	Optional	Scan Stop	MHz	The maximum frequency to be used by the peak finding algorithm. See startfreq description for more information.

Response: JSON formatted response including the following fields:

JSON field	Unit	Description
rc	-	The result code indicating whether the measurement was successful. See Section 4 for description of result codes.
msg	-	A message containing information about the request e.g. a description of any errors.
frequencies	MHz	An array containing the list of detected frequencies.
powers	mdBm	An array of associated power measurements for each of the detected frequencies.

Examples:

```
http://localhost:8002/analysis/wm?ip=WA000123.local&mod
e=peak
```

In this example, the wavelength meter takes a single scan using the WaveAnalyzer 1500S with serial number WA000123. From this scan it calculates center frequencies using `peak` mode. Default values for threshold and excursion are used, and the algorithm is performed over the full scan range.

```
{"rc":0,"msg":"Success","frequencies":[192299688,
193500312],"powers":[-8559,-7053]}
```

```
http://localhost:8002/analysis/wm?startfreq=193000000&s
topfreq=194000000&width=20000&averages=16
```

In this example, the wavelength meter takes a single scan using the WaveAnalyzer 1500S with serial number WA000123 via its mDNS name WA000123.local. The server takes an average of 16 scans and looks for peaks within a spectral range between 193-194 THz. Power values are calculated by integrating over a 20 GHz window centered at the peak of each detected frequency. The reported center frequency is calculated using the `moments` mode.

```
{"rc":0,"msg":"Success","frequencies":[193500087],"powe
rs":[-1580]}
```

5.4 Multi Channel Analysis (Wavelength Division Multiplexing Measurements)

Description: This method is used to perform a Multi Channel Analysis (Wavelength Division Multiplexing or WDM) analysis on the trace data. It returns channelized frequency, power and OSNR measurements.

API method: wdm

URL: http://<server address>/analysis/wdm?<input parameters>

Input Parameters:

Input Parameter	Type	Default	Unit	Description
startchannelfreq	Optional	191200000	MHz	The center frequency of the first channel. Use in conjunction with width to define the WDM channel plan.
width	Optional	50000	MHz	The width of a channel.
mode	Optional	moments	-	The method used to calculate the center frequency. Available options include: `moments` – The center frequency is given as the spectral centroid, or 'center of mass' of the spectrum within each valid channel width. `peak` – The center frequency is given by finding the point with the highest power value within each valid channel width.
threshold	Optional	-30000	mdBm	The minimum power required to be considered a valid channel.
offsets	Optional	*6 point measurement with auto-generated points (See Note 1 below)*	MHz	A comma-separated list of frequency offsets. The offsets are applied relative to the channel center frequency and specify the power and noise bandwidths used for the OSNR calculation. For 3-point or 6-point OSNR algorithms, the user must supply three or six frequencies respectively. See Note 1 below
startfreq	Optional	Scan Start	MHz	The minimum frequency to be analyzed. This is used in conjunction with stopfreq in order to specify a frequency range over which the WDM algorithm operates.
stopfreq	Optional	Scan Stop	MHz	The maximum frequency to be analyzed. See startfreq description for more information.

Note 1: In case the offsets parameter is not specified, a set of 6-point OSNR offsets are auto-generated based on the specified channel width. For channel width C, these offsets are generated using the following:

Offset 1: $-\frac{1}{2}\mathrm{C} - 6.25\ \mathrm{GHz}$

Offset 2: $-\frac{1}{2}\mathrm{C} + 6.25\ \mathrm{GHz}$

Offset 3: $-\frac{1}{4}\mathrm{C}$

Offset 4: $+\frac{1}{4}\mathrm{C}$

Offset 5: $+\frac{1}{2}\mathrm{C} - 6.25\ \mathrm{GHz}$

Offset 6: $+\frac{1}{2}\mathrm{C} + 6.25\ \mathrm{GHz}$

Response: JSON formatted response including the following fields:

JSON field	Unit	Description
rc	-	The result code indicating whether the measurement was successful. See Section 4 for description of result codes.
msg	-	A message containing information about the request e.g. a description of any errors.
channels	-	An array of channel numbers corresponding to each detected channel.
frequencies	MHz	An array of measured frequencies corresponding to each detected channel.
powers	mdBm	An array of measured powers corresponding to each detected channel.
osnr	mdB	An array of calculated OSNR values corresponding to each detected channel.

Examples:

```
http://localhost:8002/analysis/wdm
```

In this example, the server connects to a WaveAnalyzer with default IP 192.168.2.8. A fresh measurement is taken and a Multi Channel Analysis is performed across the whole spectrum using the default ITU 50 channel plan. OSNR values are calculated using a 6-point method with offsets automatically generated to be: -31.25 GHz, -18.75 GHz, -12.5 GHz, 12.5 GHz, 18.75 GHz, and 31.25 GHz. The result (lengthy sections omitted):

```
{"rc":0,"msg":"Success","channels":[45,…,58],"frequenci
es":[193399731,…,194050026],"powers":[-16890,…,-
16600],"osnr":[30250,…,29640]}
```

```
http://localhost:8002/analysis/wdm?threshold=-
20000&offsets=-18000,-15000,-12500,12500,15000,18000
```

This example performs a WDM analysis using the default ITU 50 channel plan. Channels with signal powers below -20 dBm are rejected. OSNR values are calculated using the 6-point method using custom offsets: -18 GHz, -15 GHz, -12.5 GHz, 12.5 GHz, 15 GHz, 18 GHz. The result (lengthy sections omitted):

```
{"rc":0,"msg":"Success","channels":[45,…,58],"frequenci
es":[193399772,…,194050062],"powers":[-16760,…,-
16500],"osnr":[27040,…,32850]}
```

```
http://localhost:8002/analysis/wdm?startfreq=193300000&
stopfreq=193600000&width=100000&offsets=-30000,0,30000
```

This example performs a WDM analysis between frequencies 193.3 THz and 193.4 THz. It uses a channel spacing of 100 GHz conforming to the ITU-100 channel plan. OSNR values are calculated using a 3-point method, with noise values specified at +/-30 GHz away from the channel center frequency. The result:

```
{"rc":0,"msg":"Success","channels":[23,24],"frequencies
":[193400111,193501873],"powers":[-15360,-
14150],"osnr":[29360,31770]}
```

5.5 Power Integration Measurements

Description: This method is used to calculate the enclosed power within a specified frequency range.

API method: power

URL: http://<server address>/analysis/power?<input parameters>

Input Parameters:

Input Parameter	Type	Default	Unit	Description
startfreq	Optional	Scan Start	MHz	The start frequency of the window over which power will be integrated. Used in conjunction with stopfreq.
stopfreq	Optional	Scan Stop	MHz	The stop frequency of the window over which power will be integrated.

Response: JSON formatted response including the following fields:

JSON field	Unit	Description
power	mdBm	The power contained within the integration window.
rc	-	Result code indicating whether the measurement was successful. See Section 4**Error! Reference source not found.** for description of result codes.
msg	-	A message containing information about the request, e.g. a description of any errors.

Example:

```
http://localhost:8002/analysis/power?startfreq=19330000
0&stopfreq=193425000
```

Calculates the total power contained within 193.3 THz and 193.425 THz. Result:

```
{"rc":0,"msg":"Success","power":-16730}
```

```
http://localhost:8002/analysis/power
```

Calculates the total power across the entire measured scan. Result:

```
{"rc":0,"msg":"Success","power":-5590}
```

5.6 Server Status

Description: Queries the status of the WaveAnalyzer Analysis Server. This is useful for when an automated test system programmatically starts the server application and must query its status.

API method: serverstatus

URL: http://<server address>/analysis/serverstatus>

Response: JSON formatted response including the following fields:

JSON field	Description
status	Returns true if the server is running. If the server is not running, then this command will not return with a response.

Example:

```
http://localhost:8002/analysis/serverstatus
```

```
{"status":true}
```

For more information on the WaveAnalyzer please contact your local WaveAnalyzer sales representative or visit http://www.finisar.com/instruments.

FINISAR

1389 Moffett Park Drive
Sunnyvale, CA 94089, USA
Tel.: +1-408-548-1000
Fax: +1-408-541-6138
waveanalyzer@finisar.com
http://www.finisar.com/instruments

©2015-2018 Finisar Corporation. All rights reserved. Finisar is a registered trademark.