

Polycom[®] Real-time Signal Level and Microphone Gating Information from SoundStructure

This document describes how to get real-time signal meter and microphone gating information, collectively called meter information, from SoundStructure devices over RS-232 or Ethernet to a host computer or control system. Real-time meter information may be useful for camera positioning applications that require real-time microphone gating information or with in-room diagnostic tests such as those that verify that all microphones are active.

Introduction

There are many parameters, such as mute and fader, within SoundStructure devices that may have their values set and values queried via the SoundStructure API. In addition, SoundStructure devices can send meter information to a control system or connected computer via the SoundStructure's Ethernet and RS-232 ports. This meter data can include signal level information and microphone gating status.

There are three steps involved with getting meter information from a SoundStructure device:

1. Registering for the desired meter data
2. Receiving the meter data reception and processing/parsing the information
3. Unregistering for the meter data when the data is no longer needed

These steps are described in the following sections. To illustrate the meter API, we will use an example of registering for preprocessing signal levels and automixer gate status on the two virtual channels "Mic 1" and "Mic 2".

Meter Registration

All meter data must be 'registered' with SoundStructure before the data is available. Registering for meters requires sending a mtrreg command to the SoundStructure device via any control port (RS-232 or Ethernet). SoundStructure supports multiple simultaneous control connections over Ethernet and a single session over RS-232.

Meters registered on one Ethernet connection will not be sent to another Ethernet connection (or to RS-232) which means that any connection that wants meter information must register for that information regardless of what's been registered on other ports. All meters registered on the RS-232 port will stay enabled until they are unregistered. Meters registered on an Ethernet connection will stay enabled until the meters are unregistered or the connection is terminated.

Meters are registered with the mtrreg command using the following syntax.

mtrreg <meter name> <virtual channel name> <virtual channel index>

where the arguments are described in the table below.

Argument	Description
<meter name>	The name of the meter to be registered. Can be level_pre, level_triune_cr_post, level_triune_sr_post, level_triune_raw_post, am_gate, am_camera_gate
<virtual channel name>	The name of the virtual channel on which the meter will be registered.
<virtual channel index>	The index of the physical channel within the virtual channel on which the meter will be registered. This value will be 1 for mono virtual channels. For stereo virtual channels, use 1 for the left channel and 2 for the right.

Supported meter names include level_pre, level_post, and am_gate and am_camera_gate. A list of these meter names is given at the end of this document.

The meter registration commands for our example are:

```
mtrreg am_gate "Mic 1" 1
mtrreg am_gate "Mic 2" 1
mtrreg level_pre "Mic 1" 1
mtrreg level_pre "Mic 2" 1
```

For each of the mtrreg commands, SoundStructure will send acknowledgments that look exactly like the commands. In our example, we would receive the following acknowledgments.

```
mtrreg am_gate "Mic 1" 1
mtrreg am_gate "Mic 2" 1
mtrreg level_pre "Mic 1" 1
mtrreg level_pre "Mic 2" 1
```

Meter Data

As soon as the first meter registration command is sent, the SoundStructure device will start sending meter data with the mtrdata command response that includes the data for all currently registered meters sent in a single mtrdata command. Since some mtrdata commands might be sent before all the mtrreg commands have been processed, it is required to either ignore the first several mtrdata commands or be able to process the mtrdata commands that contain less than the full amount of data.

Meter data appears in the order in which it was registered and the mtrdata command has the following syntax.

```
mtrdata <escaped data>
```

There is a single space between the mtrdata command name and the first byte of the escaped data. The escaped data consists of the following data processed by an escaping algorithm.

```
<meter 1><meter 2>...<meter N>
```

Each meter item corresponds to a meter that was registered, and appears in the order in which they were registered. Most meters have data that consists of a single byte, but some have two- or four- byte meter data.

For our example, the data in the meter command would be:

```
<gate for Mic 1><gate for Mic 2><level for Mic 1><level for Mic 2>
```

Because certain characters may not appear in commands or have special meanings in commands, these characters must be escaped if they occur in the meter data. To escape one of these characters, SoundStructure replaces the offending byte with an escape byte (0x1B) followed by the data byte plus 32 (0x20). To unescape the data, your program must look for any occurrence of the escape byte, discard it, and subtract 32 from the next byte.

Example code in C for the unescape algorithm is given below.

```
int unescape_data(unsigned char *escaped_data, int num_escaped_bytes, unsigned char *unescaped_data)
{
    int num_unescaped_bytes;
    int i;

    for (i = 0, num_unescaped_bytes = 0;
         i < num_escaped_bytes;
         i++, num_unescaped_bytes++) {
        if (escaped_data[i] == 0x1B) {
            i++;
            unescaped_data[num_unescaped_bytes] = escaped_data[i] - 32;
        }
        else {
            unescaped_data[num_unescaped_bytes] = escaped_data[i];
        }
    }
    return num_unescaped_bytes;
}
```

Meter Data Period

The rate that the control system receives the data may be controlled with the global parameter `sys_meter_period`. The meter period parameter corresponds approximately to the number of milliseconds between meter data values. Setting this value to 1 will make the system send the data as fast as possible and setting the value to 10000 will set the data rate to be every 10 seconds. The syntax for setting the meter period is shown below

```
set sys_meter_period 100
```

and the system will respond with the command response:

```
val sys_meter_period 100
```

when the value is set to 100.

Please note that the `sys_meter_period` parameter is a global parameter – setting this value will affect how the meters are displayed in SoundStructure Studio. SoundStructure Studio will also set this value if SoundStructure Studio is connected at the same time as the control system.

Meter Unregistration

Meters are unregistered using the `mtrunreg` command. The meters may be unregistered individually using the following syntax.

```
mtrunreg <meter name> <virtual channel name> <virtual channel index>
```

However, it is easier to unregister all the meters at once using the following shortcut syntax.

```
mtrunreg all
```

In both cases, SoundStructure will respond with an acknowledgment that is exactly the same as the command. If only certain meters are unregistered, remember the remaining meter information will still be sent with the mtrdata command in the original order except now the meter channels that have been unregistered are no longer in the data stream.

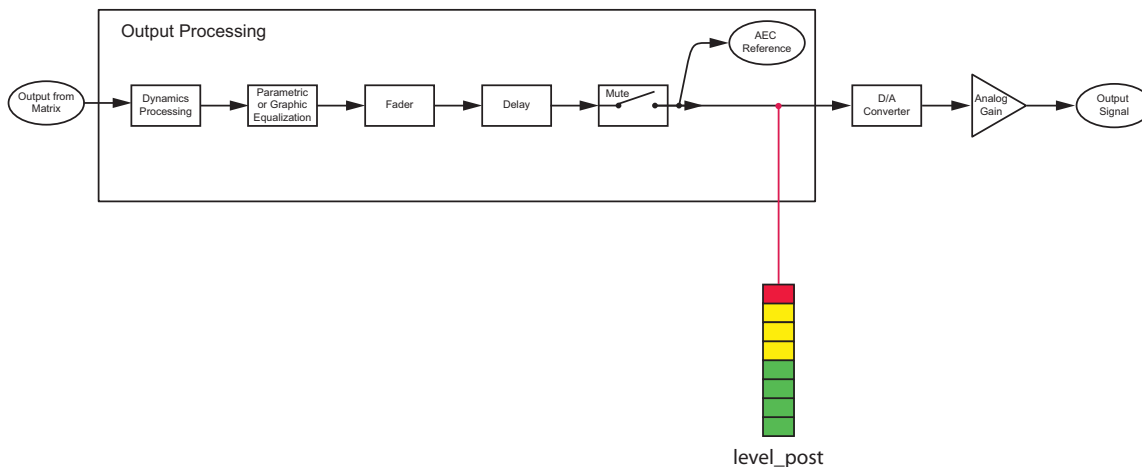
List of Meters

level_post

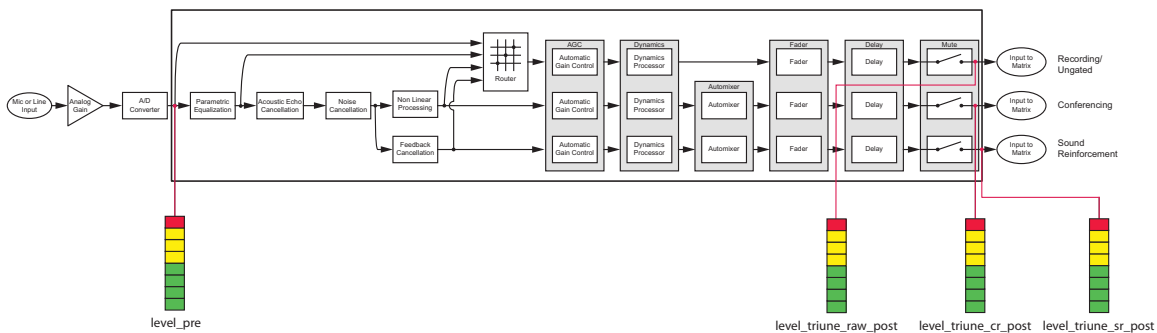
This meter is used with output signals from SoundStructure and the meter sends a 1-byte value for each data sample with the 8-bit value mapped so that 135 = -100 dB, and 255 = +20 dB. This meter reports the signal level (in dB) after all the digital signal processing.

To receive the post processing level information for input channels, use the level_triune_cr_post, level_triune_sr_post, and level_triune_raw_post meters.

The location of the level_post meter in the output processing path is shown in the following figure.



The location of the input signal meters in the input processing path is shown in the following figure.



level_pre

This meter sends a 1-byte value for each piece of data with the 8-bit value mapped so that 135 = -100 dB, and 255 = +20 dB. This meter gives the signal level (in dB) of the signal before any digital signal processing and is positioned just after the A/D converter.

level_triune_cr_post

This meter sends a 1-byte value for each piece of data with the 8-bit value mapped so that 135 = -100 dB, and 255 = +20 dB. This meter reports the signal level (in dB) of the conferencing triune signal after processing. The conferencing triune corresponds to the "Conferencing" gated type or "blue" crosspoint as used in the matrix.

level_triune_sr_post

This meter sends a 1-byte value for each piece of data with the 8-bit value mapped so that 135 = -100 dB, and 255 = +20 dB. This meter reports the signal level (in dB) of the sound reinforcement triune signal after processing. The sound reinforcement triune corresponds to the "Snd Reinforcement" gated type or "turquoise" crosspoint as used in the matrix.

level_triune_raw_post

This meter sends a 1-byte value for each piece of data with the 8-bit value mapped so that 135 = -100 dB, and 255 = +20 dB. This meter reports the signal level (in dB) of the ungated triune signal after processing. The ungated triune corresponds to the "None" gated type or "white" crosspoint as used in the matrix.

am_gate

This meter sends a 1-byte value for each piece of data with the 8-bit value mapped so that the value of 32 corresponds to a microphone gated off and a value of 33 corresponds to a microphone gated on (active).

am_camera_gate

This meter sends a 1-byte value for each piece of data with the 8-bit value mapped so that the value of 32 corresponds to a microphone camera gated off and a value of 33 corresponds to a microphone camera gated on (active). The configuration of the amount of time a microphone must be on to be considered "camera gated" is specified by the "Camera Act. Time" control on the Automixer page with SoundStructure Studio.