



Worldwide Headquarters  
3455 Bethesda Road  
Stouffville, Ontario  
L4A 3Z4, Canada

+1 (905) 770-0055  
hello@xilica.com  
www.xilica.com

## Networked Audio Solutions

### Third-Party Control Protocol: User Manual

*Version 1.1*

#### Getting Help

For assistance, contact our Global Customer Solutions Team via email at [support@xilica.com](mailto:support@xilica.com). If you would like to speak with a specific Customer Solutions Engineer, visit [www.xilica.com/partners/](http://www.xilica.com/partners/) to find your local Xilica support team, or contact the distributor in your territory. You may also call your nearest office:

North America/Rest of World: +1 (905) 770-0055 (inc. out-of-hours support)

Europe: +31 29940 1100

China: +86 13602279067



## Introduction

Since the introduction of Xilica's Neutrino and Uno DSPs, user feedback has revealed that the protocol is too complex for integration with other third-party controllers. Numerous minor modifications were done in the interim to assist users in utilizing the Neutrino/Uno Protocol. Although the Neutrino/Uno Protocol itself is very generic and fully capable, it has become increasingly important to define a new protocol that better suits common third-party controllers. Our new third-party control protocol was designed with this in mind, enabling existing and future Xilica devices to be easily controlled by popular third-party controllers from Crestron, AMX and others.

## Overview

The new third-party control protocol is transport-independent. This means the syntax is the same whether the third-party controller is connecting using Serial (RS232/USB) or Ethernet connection. However, some functionality (Subscriptions and Password Protection) is not available when using the protocol in Serial mode.

For Ethernet connection, the user should send out messages using TCP port #10007. The server will Respond to the message using the same TCP connection. A keep-alive message must be sent over this TCP connection every 60 seconds, otherwise, the server end will disconnect the TCP connection and all subscriptions associated with the connection will end.

Users also have the choice of using UDP port #10008 to listen to Subscription messages from the device. The user can select whether a particular parameter sends out its change via TCP Unicast or UDP Broadcast when issuing a subscription command. If left unspecified, by default a parameter will notify via TCP Unicast. A separate TCP connection is mandatory for status update, and a keep-alive message must continuously be sent over this TCP connection even if the user chooses to use UDP Broadcast for all their interested parameters. If at any instance the TCP connection is dropped, all subscriptions and groups settings in the device must be reconfigured again.

## Syntax

The third-party controller string is composed with human readable ASCII characters. Each field is separated by one single white space. Using more than one white space in between fields will result in command parsing error. A carriage-return (<CR>) is sent to mark the end of the message.



Fields enclosed in square brackets are dependent on the command. Refer to the Commands List section on Page 6,7, & 8 for a list of all commands, their detail usage and examples.

COMMAND	1 white space	[CONTROL OBJECT/GROUP]	1 white space	[DATA]	<CR>
---------	---------------	------------------------	---------------	--------	------

The CONTROL OBJECT is a string of up to 32 characters assigned by the user in software for individual parameters. It can contain any readable ASCII characters except double quotes. However, the first character cannot be a dollar-sign (\$) because a preceding '\$' is used to distinguish between a CONTROL OBJECT with a CONTROL GROUP.

A CONTROL GROUP is a string of up to 32 characters created using the CREATE command for use as a group name. It can also contain any readable ASCII characters except double quotes. The first character of the CONTROL GROUP will always begin with a '\$' to denote it as a group name.

For CONTROL OBJECT/GROUP, if any white spaces are used as part of the string, then it must be encapsulated by double quotes. In addition, note that both COMMAND and CONTROL OBJECT/GROUP are case-sensitive.

DATA can be either:

- a number (positive, negative, floating point, integer represented in ASCII)
- a string (must always be inside double quotes, case-sensitive)
- a Boolean (TRUE or FALSE, case-sensitive)

Refer to the Commands List section for details on the data type accepted by each command.

## Responses

The device end will respond to a third-party control command regardless if it is correct or not. If no response is received, it is likely an indication of a connection problem. All response messages from the device will end with a carriage-return (<CR>).

If an invalid command is sent, the last encountered Error Code will be returned as:  
ERROR=<ERROR CODE><CR>



For GET or GETRAW command, the response will be:  
<CONTROL OBJECT>=<DATA><CR>

For all other commands, the device will return:  
OK<CR>

## Subscriptions

The external controller can subscribe to control objects to get a notification for any data changes on the subscribed objects.

To subscribe/unsubscribe to a control object, simply send the command:  
SUBSCRIBE <CONTROL OBJECT> [TCP/UDP] <CR>  
UNSUBSCRIBE <CONTROL OBJECT> <CR>

The notification will then be automatically sent to the external control system via TCP Unicast or UDP Broadcast as specified in the command. The notification string received by the external controller will be:  
#<CONTROL OBJECT>=<DATA><CR>

The string is similar to a GET command, with a # character added in front to distinguish between an explicit read or a notification.

The interval in which the device sent out notifications is global for all subscribed control objects, it can be configured by:  
INTERVAL <TIME in milliseconds> <CR>

## Control Groups

Control groups allow a user to control multiple parameters at once using a single command. The user must first create a group by:  
CREATE <CONTROL GROUP> <CR>

After a group is created, individual control objects can join or leave the group by:  
JOIN <CONTROL GROUP> <CONTROL OBJECT> <CR>  
LEAVE <CONTROL GROUP> <CONTROL OBJECT> <CR>



Caution must be used when adding parameters to a group to ensure that the parameters are all of the same type and support the same commands.

When a group is no longer used, resources can be freed up by:  
REMOVE <CONTROL GROUP> <CR>

Similar to subscription, Control Groups are persistent for the duration of the active connection only. When a connection is lost, the groups must be recreated again.

### **Password Protection**

If a device is protected with a password, then the user must unlock the device first before sending any commands. The authentication persists only for the duration of the connection, so if a TCP disconnection occurs, the user will have to unlock the device again.

To unlock the device, send the following command:  
LOGIN <PASSWORD> <CR>

The password used in the command is the same password setup in software.

### **Verbose/Simple Mode**

<Future Implementation - mainly used to configure the amount of details for Responses. This will help some external controller parsing the response more easily>



## Commands List

<b>SET &lt;CONTROL OBJECT/GROUP&gt; &lt;DATA - number/string/Boolean&gt;</b>		
Examples	SET gain1 -3.2	Set "gain1" to -3.2 dB
	SET polarity1 TRUE	Set "polarity1" to ON position
	SET filter1 "Butterworth"	Set "filter1" to Butterworth Filter
	SET \$group1 -15.7	Set all parameters in group1 to -15.7dB

<b>SETRAW &lt;CONTROL OBJECT/GROUP&gt; &lt;DATA - number &gt;</b>		
Examples	SETRAW gain1 -3200	Set "gain1" to -3.2 dB
	SETRAW polarity1 1	Set "polarity1" to ON position
	SETRAW filter1 1	Set "filter1" to Butterworth Filter
	SETRAW \$group1 1000	Set all parameters in group1 to +1.0dB

<b>GET &lt;CONTROL OBJECT/GROUP&gt;</b>		
Example	GET EQslope	Get "EQslope" formatted value
	GET \$group1	Get formatted value for all parameters in group1

<b>GETRAW &lt;CONTROL OBJECT/GROUP&gt;</b>		
Example	GETRAW EQslope	Get "EQslope" raw value
	GETRAW \$group1	Get raw value for all parameters in group1

<b>INC &lt;CONTROL OBJECT/GROUP&gt; &lt;DATA - number&gt;</b>		
Example	INC fader3 0.5	Increase "fader3" by 0.5 dB
	INC \$group1 1	Increase all parameters in group1 by 1dB



<b>INCRAW &lt;CONTROL OBJECT/GROUP&gt; &lt;DATA - number&gt;</b>		
Example	INC fader3 500	Increase "fader3" by 0.5 dB
	INC \$group1 1000	Increase all parameters in group1 by 1dB

<b>DEC &lt;CONTROL OBJECT/GROUP&gt; &lt;DATA - number&gt;</b>		
Example	DEC fader3 0.5	Decrease "fader3" by 0.5 dB
	DEC \$group1 1	Decrease all parameters in group1 by 1dB

<b>DECRAW &lt;CONTROL OBJECT/GROUP&gt; &lt;DATA - number&gt;</b>		
Example	DEC fader3 500	Decrease "fader3" by 0.5 dB
	DEC \$group1 1000	Decrease all parameters in group1 by 1dB

<b>TOGGLE &lt;CONTROL OBJECT/GROUP&gt;</b>		
Example	TOGGLE mute1	Toggle "mute1" state
	TOGGLE \$group2	Toggle all parameters in group2

<b>PRESET &lt;DATA - number/string&gt;</b>		
Example	PRESET 4	Recall preset #4
	PRESET "preset name"	Recall preset with name "preset name"

<b>SUBSCRIBE &lt;CONTROL OBJECT/GROUP&gt; &lt;DATA - string&gt; *</b>		
Example	SUBSCRIBE meter6	Subscribe to "meter6" via TCP Unicast
	SUBSCRIBE meter6 "TCP"	Subscribe to "meter6" via TCP Unicast
	SUBSCRIBE meter6 "UDP"	Subscribe to "meter6" via UDP Broadcast

\* <DATA - string> is optional, TCP Unicast will be used by default

<b>UNSUBSCRIBE &lt;CONTROL OBJECT/GROUP&gt;</b>		
Example	UNSUBSCRIBE meter6	Unsubscribe "meter6"
<b>KEEPALIVE</b>		
Example	KEEPALIVE	No operation. Can be used by external controller to keep the TCP connection alive.

<b>INTERVAL &lt;DATA - number&gt;</b>		
Example	INTERVAL 100	<p>Set subscription interval to minimum 100ms.</p> <p>* Subscription data could be delayed longer than the specified interval due to CPU usage, but it is guaranteed to wait for the configured interval time before attempting to send out subscription data.</p> <p>* The minimum value is 100 ms.</p>

<b>LOGIN &lt;DATA - string&gt;</b>		
Example	LOGIN "password"	Login for external control with "password"

<b>REBOOT</b>		
Example	REBOOT	Remotely reboot device

<b>REFRESH</b>		
Example	REFRESH	Get formatted data value for all control objects

<b>CREATE &lt;CONTROL GROUP&gt;</b>		
Example	CREATE group1	<p>Create a group with the name "group1 "</p> <p>This is the only exception where CONTROL GROUP does not require a '\$' sign in the syntax because the '\$' sign will be automatically added when the group is created.</p>

<b>REMOVE &lt;CONTROL GROUP&gt;</b>		
Example	REMOVE \$group1	Remove the group with name "group1 "



<b>JOIN &lt;CONTROL GROUP&gt; &lt;DATA - string&gt;</b>		
Example	JOIN \$group1 "gain1"	"gain1" will join group1

<b>LEAVE &lt;CONTROL GROUP&gt; &lt;DATA - string&gt;</b>		
Example	LEAVE \$group2 "mute2"	"mute2" will leave group2

### Data String

<b>Filter Type</b>
Butterworth, LR, Bessel
<b>Filter Slope</b>
6db/Oct, 12db/Oct, 18db/Oct, 24db/Oct, 30db/Oct, 36db/Oct, 42db/Oct, 48db/Oct
<b>AFS Sensitivity</b>
Very Low, Low, Medium, High, Very High
<b>AFS Type</b>
Dynamic, Fixed
<b>Control Ramp Type</b>
Linear, Log, Audio



## Error Codes

Error Code	Description
101	Invalid Command
102	Bad Arguments
103	Invalid Data Format
104	Control Object Not Found
105	Parameter Not Found
106	Data Value Not Found
107	Max Subscription Reached
108	Password Error
109	Not Yet Login
110	Command Not Supported for Control Object
111	Invalid Group Name
112	Max Control Group Reached
113	Max Control Object in Group Reached
114	Object Already in Group
115	Object Not in Group
116	Conflicting With Other Objects in Group
117	Invalid Preset #
118	Invalid Preset Name

