Tektronix®

Multitone Plug-in Application

Programmer Manual

077-1239-01

Tektronix®

Multitone Plug-in Application

Programmer Manual

www.tek.com

077-1239-01

Copyright © Tektronix. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

SourceXpress ® is a registered trademark of Tektronix, Inc.

Contacting Tektronix

Tektronix, Inc.
14150 SW Karl Braun Drive
P.O. Box 500
Beaverton, OR 97077
USA

For product information, sales, service, and technical support:

- In North America, call 1-800-833-9200.
- Worldwide, visit www.tek.com to find contacts in your area.

Table of Contents

Getting Started

Syntax and Commands

Status and Events

Getting Started

Introduction

This programmer manual provides information on how to use commands for remotely controlling the Multitone plug-in application.

The Multitone plug-in can be installed in either the SourceXpress software application or an AWG70000 series arbitrary waveform generator.

Communication with the plug-in is through the SourceXpress programmatic interface or the AWG70000 series instrument's programmatic interface. Using a single VISA or raw socket session, it is possible to communicate with both SourceXpress and AWG70000 series instruments.

For information on the Remote Control, GPIB Parameters, LAN Parameters, Connecting to the Instrument using GPIB, and Setting up GPIB Communication, refer to the *AWG70000A series Arbitrary Waveform Generators Programmer Manual*.

Documentation

In addition to this Multitone Programmer Guide, the following documentation is included with this application:

- Multitone Help. The help provides in-depth operation and user interface help.
- Multitone User Manual (PDF). This is adapted from the Multitone help system.

Syntax and Commands

Command Syntax

For information on the Syntax Overview, Command and Query Structure, Clearing the Instrument, Command Entry, Parameter Types, SCPI Commands and Queries, refer to the *AWG70000A Series Arbitrary Waveform Generators Programmer Manual*.

Command Groups

Common commands

Table 2-1: Common commands and their descriptions

Command	Description
WPLugin:ACTive	This command sets or returns the active waveform plug-in.
MTONe:TYPE	This command sets or returns the type of compilation (Tones or Chirp) that occurs.
MTONe:RESet	This command resets the Multitone plug-in to its default values.

Chirp commands

Table 2-2: Chirp commands and their descriptions

Command	Description
MTONe:CHIRp:FSWeep	This command sets or returns the Frequency Sweep of the Chirp waveform.
MTONe:CHIRp:HIGH	This command sets or returns the High Frequency of the Chirp waveform in Hz.
MTONe:CHIRp:LOW	This command sets or returns the Low Frequency of the Chirp waveform in Hz.
MTONe:CHIRp:SRATe	This command sets or returns the Sweep Rate of the Chirp waveform in Hz/µs.
MTONe:CHIRp:STIMe	This command sets or returns the Sweep Time of the Chirp waveform in seconds.

Tones commands

Table 2-3: Tones commands and their descriptions

Command	Description
MTONe:TONes:STARt	This command sets or returns the Start Frequency of the Tone waveform in Hz.
MTONe:TONes:END	This command sets or returns the End Frequency of the Tone waveform in Hz.
MTONe:TONes:PHASe	This command sets or returns the Phase for the Tone waveform.
MTONe:TONes:PHASe:UDEFined	This command sets or returns the User Defined Phase value in degrees for the Tone waveform.

Table 2-3: Tones commands and their descriptions (cont.)

Command	Description
MTONe:TONes:SPACing	This command enables Tones Spacing and sets the of the Spacing value in Hz. The query form returns the Spacing setting.
MTONe:TONes:NTONes	This command enables the Tone Count and sets the number. The query form returns the Tone Count setting.
MTONe:TONes:NOTCh:ENABle	This command enables if notches will be added during the Tones compile. The query form returns the current value.
MTONe:TONes:NOTCh:ADD	This command adds a new notch to the Notch table with the given Start and End frequencies.
MTONe:TONes:NOTCh[n]:DELete	This command deletes the notch at the specified index or all notches.
MTONe:TONes:NOTCh:COUNt?	This command returns the number of notches in the notch table.
MTONe:TONes:NOTCh[n]	This command sets both the Start and End frequency of the notch at the specified index.
MTONe:TONes:NOTCh[n]:STARt	This command sets or returns the Start frequency of the notch at the specified index.
MTONe:TONes:NOTCh[n]:END	This command sets or returns the End frequency of the notch at the specified index.

Compile commands

Table 2-4: Compile commands and their descriptions

Command	Description
MTONe:COMPile:NAMe	This command sets the name of the signal that is to be compiled.
MTONe:COMPile:PLAY	This command enables the “Play after assign” setting in the Compile Settings. The query form returns the current status.
MTONe:COMPile:CASSign	Sets or returns the state (enabled or disabled) to compile the waveform and immediately assign it to a specified channel (enabled) or just compile the waveform (disabled).
MTONe:COMPile:CHANnel	This command sets or returns which channel the signal will be assigned to upon compile.

Table 2-4: Compile commands and their descriptions (cont.)

Command	Description
MTONe:COMPile:SRATe	This command sets or returns the sampling rate of the signal in Hz.
MTONe:COMPile:SRATe:AUTO	This command sets or returns if the sampling rate will be automatically calculated at compile time.
MTONe:COMPile:CORRection:APPLy	This command sets or returns if a correction file is to be applied during compile.
MTONe:COMPile:CORRection:PATH	This command sets or returns the correction file's filepath and filename to use during compile.
MTONe:COMPile	This command compiles and generates a waveform using the Multitone plug-in compile settings.
MTONe:COMPile:CANCel	This command cancels a compilation currently in progress.

Command Descriptions

WPLugin:ACTive

This command sets or returns the active waveform plug-in. To use the Multitone commands in this document, the active waveform plug-in must be set to Multitone.

Group Control

Syntax

```
WPLugin:ACTive <plug-in_name>
WPLugin:ACTive?
```

Arguments <plugin_name>::=<string>

A single string representing the waveform plug-in name.

“Multitone” is the proper string to activate the Multitone plug-in.

Returns <plugin_name>::=<string>

A single string representing the active waveform plug-in.

Examples `WPLUGIN:ACTIVE "Multitone"` sets the Multitone plug-in as the active plug-in.

`WPLUGIN:ACTIVE?` might return `"Multitone"` indicating Multitone is currently the active waveform plug-in.

MTONe:TYPE

This command sets or returns the type of compilation (Tones or Chirp) that occurs.

Conditions The active plug-in must be Multitone.

Group Control

Syntax `MTONe:TYPE {TONes|CHIRp}`

Related Commands WPLugin:ACTive
MTONe:COMPile

Arguments TONes – Sets Tones mode as active

CHIRp – Sets Chirp mode as active

Returns TON – Tones mode is currently the selected type

CHIR – Chirp mode is currently the selected type

Examples `MTONE:TYPE CHIRP` sets the Multitone type to Chirp.

`MTONE:TYPE ?` might return `TON`, indicating the Multitone type is set to Tones.

MTONe:RESet (No Query Form)

This command resets the Multitone plug-in to its default values.

Conditions The active plug-in must be Multitone.

Group Control

Syntax `MTONe:RESet`

Related Commands WPLugin:ACTive

Examples `MTONE:RESET` returns the Multitone plug-in to its default values.

MTONe:CHIRp:FSWeep

This command sets or returns the Frequency Sweep of the Chirp waveform to create a waveform that sweeps from the low to high frequency settings or from the high to low frequency settings.

Conditions The active plug-in must be Multitone.

Group Chirp

Syntax

```
MTONe:CHIRp:FSWeep {LHIGh|HLOW}
MTONe:CHIRp:FSWeep?
```

Related Commands WPLugin:ACTive

Arguments LHIGh sets the Chirp waveform to sweep from Low to High frequencies.

HLOW sets the Chirp waveform to sweep from High to Low frequencies.

Returns LHIG = sweep from Low to High frequencies

HLOW = sweep from High to Low frequencies

Examples `MTONE:CHIRP:FSWEEP LHIGh` sets the Chirp Frequency Sweep setting to Low to High.

`MTONE:CHIRP:FSWEEP?` might return `HLOW`, indicating the Frequency Sweep is set to High to Low.

MTONe:CHIRp:HIGH

This command sets or returns the High Frequency of the Chirp waveform in Hz.

Conditions The active plug-in must be Multitone.

Group Chirp

Syntax

```
MTONe:CHIRp:HIGH <NR3>
MTONe:CHIRp:HIGH?
```

Related Commands WPLugin:ACTive
MTONe:CHIRp:LOW

Arguments A single <NRf> value

Returns A single <NR3> value

Examples `MTONE:CHIRP:HIGH 2E9` sets the Chirp High Frequency to 2 GHz.

`MTONE:CHIRP:HIGH?` might return `3.7000000000E+9`, indicating the High Frequency is set to 3.7 GHz.

MTONe:CHIRp:LOW

This command sets or returns the Low Frequency of the Chirp waveform in Hz.

Conditions The active plug-in must be Multitone.

Group Chirp

Syntax

```
MTONe:CHIRp:LOW <NRf>
MTONe:CHIRp:LOW?
```

Related Commands WPLugin:ACTive
MTONe:CHIRp:HIGH

Arguments A single <NRf> value

Returns A single <NR3> value

Examples `MTONE:CHIRP:LOW 1E9` sets the Chirp Low Frequency to 1 GHz.

`MTONE:CHIRP:LOW?` might return `2.3000000000E+9`, indicating the Low Frequency is set to 2.3 GHz.

MTONe:CHIRp:SRATe

This command sets or returns the Sweep Rate of the Chirp waveform in Hz/µs.

Conditions The active plug-in must be Multitone.

Group Chirp

Syntax

```
MTONe:CHIRp:SRATe <NRf>
MTONe:CHIRp:SRATe?
```

Related Commands WPLugin:ACTive

Arguments A single <NRf> value

Returns A single <NR3> value

Examples `MTONE:CHIRP:SRATE 1.6E9` sets the Chirp Sweep Rate to 1.6 GHz/µs.

`MTONE:CHIRP:SRATE?` might return `2.4000000000E+9`, indicating the Chirp Sweep Rate is set to 2.4 GHz/µs.

MTONe:CHIRp:STIMe

This command sets or returns the Sweep Time of the Chirp waveform in seconds.

Conditions The active plug-in must be Multitone.

Group Chirp

Syntax

```
MTONe:CHIRp:STIMe <NRf>
MTONe:CHIRp:STIMe?
```

Related Commands WPLugin:ACTive

Arguments A single <NRf> value

Returns A single <NR3> value

Examples `MTONE:CHIRP:STIME 8E-3` sets the Chirp Sweep Time to 8 ms.

`MTONE:CHIRP:STIME?` might return `2.4000000000E-3`, indicating the Sweep Time is set to 2.4 ms.

MTONe:TONes:STARt

This command sets or returns the Start Frequency of the Tone waveform in Hz.

Conditions The active plug-in must be Multitone.

Group Tones

Syntax

```
MTONe:TONes:STARt <NRf>
MTONe:TONes:STARt?
```

Related Commands WPLugin:ACTive

Arguments A single <NRf> value

Returns A single <NR3> value

Examples `MTONE:TONES:START 1E9` sets the Tone Start Frequency to 1 GHz

`MTONE:TONES:START?` might return `2.2500000000E+9`, indicating the Tone Start Frequency is set to 2.25 GHz.

MTONe:TONes:END

This command sets or returns the End Frequency of the Tone waveform in Hz.

Conditions The active plug-in must be Multitone.

Group Tones

Syntax

```
MTONe:TONes:END <NRf>
MTONe:TONes:END?
```

Related Commands WPLugin:ACTive

Arguments A single <NRf> value

Returns A single <NR3> value

Examples `MTONE:TONES:END 2E9` sets the Tone Start Frequency to 2 GHz

`MTONE:TONES:END?` might return `3.2500000000E+9`, indicating the Tone End Frequency is set to 3.25 GHz.

MTONe:TONes:PHASe

This command sets or returns the Phase for the Tone waveform.

Conditions The active plug-in must be Multitone.

Group Tones

Syntax

```
MTONe:TONes:PHASe {RANDom|NEWMan|UDEFined}
MTONe:TONes:PHASe?
```

Related Commands WPLugin:ACTive

Arguments RANDom – The application applies a random phase shift to each of the tones.

NEWMan – The phase shift is based on the Newman phase calculations.

UDEFined – Select a specific phase shift in degrees to apply to each of the tones.

Returns RAND – Random phase shift

NEWM – Newman phase shift

UDEF – User defined phase shift

Examples `MTONE:TONES:PHASE RANDOM` sets the Phase to Random.

`MTONE:TONES:PHASE?` might return `NEWM`, indicating the Phase is set to Newman.

MTONe:TONes:PHASe:UDEFined

This command sets or returns the User Defined Phase value in degrees for the Tone waveform.

Conditions The Phase must be set to "UDEFined" for this value to be applied to the compiled waveform.

The active plug-in must be Multitone.

Group Tones

Syntax

```
MTONe:TONes:PHASe:UDEFined <NR3>
MTONe:TONes:PHASe:UDEFined?
```

Related Commands WPLugin:ACTive

MTONe:TONes:PHASe

Arguments A single <NR3> value

Range: 0 to 180

Returns A single <NR1> value

Examples `MTONE:TONES:PHASE:UDEFINED 90` sets the User Defined Phase to 90 °.

`MTONE:TONES:PHASE:UDEFINED?` might return 75.0000000000, indicating the User Defined Phase is set to 75 °.

MTONe:TONes:SPACing

This command enables Tones Spacing and sets the of the Spacing value in Hz.

The query form returns the Spacing setting.

Conditions The active plug-in must be Multitone.

Group Tones

Syntax

```
MTONe:TONes:SPACing <NRf>
MTONe:TONes:SPACing?
```

Related Commands WPLugin:ACTive
MTONe:TONes:STARt
MTONe:TONes:END

Arguments A single <NRf> value

The available range is dependent on the Start and End Frequency settings.

Returns A single <NR3> value

Examples `MTONE:TONES:SPACING 1E3` sets the Tone Spacing to 1 kHz

`MTONE:TONES:SPACING?` might return `2.1000000000E+3`, indicating the Tone Spacing is set to 2.1 kHz.

MTONe:TONes:NTONes

This command enables the Tone Count and sets the number.

The query form returns the Tone Count setting.

Conditions The active plug-in must be Multitone.

Group Tones

Syntax

```
MTONe:TONes:NTONes <NR3>
MTONe:TONes:NTONes?
```

Related Commands WPLugin:ACTive

Arguments A single <NR3> value

Returns A single <NR3> value

Examples `MTONE:TONES:NTONES 1E3` Sets the Number of Tones to 1 k.

`MTONE:TONES:NTONES?` might return `6.0000000000E+5`, indicating the Tone count is set to 600 k.

MTONe:TONes:NOTCh:ENABle

This command enables if notches will be added during the Tones compile.

The query form returns the current value.

Conditions The active plug-in must be Multitone.

Group Tones

Syntax

```
MTONe:TONes:NOTCh:ENABle {0|1|OFF|ON}
MTONe:TONes:NOTCh:ENABle?
```

Related Commands WPLugin:ACTive
MTONe:TONes:NOTCh:ADD
MTONe:TONes:NOTCh[n]:DELete

Arguments 0 or OFF disables Notches; no Notches will be applied.

1 or ON enables Notches, applying Notches during compile.

Returns A single <Boolean> value representing the setting {0|1}.

Examples `MTONE:TONES:NOTCH:ENABLE 1` will add notches to the signal during compile.

`MTONE:TONES:NOTCH:ENABLE?` might return `0` indicating that the signal will not add notches during compile.

MTONe:TONes:NOTCh:ADD (No Query Form)

This command adds a new notch to the Notch table with the given Start and End frequencies.

Conditions The active plug-in must be Multitone.

This command does not enable Notches.

Group Tones

Syntax `MTONe:TONes:NOTCh:ADD <start>,<end>`

Related Commands WPLugin:ACTive
MTONe:TONes:NOTCh:ENABle

Arguments <start> ::= <NRf>

<end> ::= <NRf>

Examples `MTONE:TONES:NOTCH:ADD 1.5E6,2.5E6` adds a Notch to the notch table (at the next available index) with a Start frequency of 1.5 MHz and an End frequency of 2.5 MHz.

MTONe:TONes:NOTCh[n]:DELete (No Query Form)

This command deletes the notch at the specified index or all notches.

NOTE. *Notches are deleted without warning.*

Conditions The active plug-in must be Multitone.

Group Tones

Syntax `MTONe:TONes:NOTCh[n]:DELete [{ALL}]`

Related Commands WPLugin:ACTive

Arguments enum ::= {ALL}
ALL indicates that all the notches in the notch table will be deleted. The enumeration is optional.

[n] is an index in the notch table <NR1> (If omitted, interpreted as 1.)

Examples `MTONE:TONES:NOTCH2:DELETE` deletes the notch at index 2.

`MTONE:TONES:NOTCH:DELETE ALL` deletes all notches.

MTONe:TONes:NOTCh:COUNt? (Query Only)

This command returns the number of notches in the notch table.

Conditions The active plug-in must be Multitone.

Group Tones

Syntax `MTONe:TONes:NOTCh:COUNt?`

Related Commands WPLugin:ACTive

Returns A single <NR1> value

Examples `MTONE:TONES:NOTCH:COUNT?` might return `2`, indicating there are two notches in the notch table.

MTONe:TONes:NOTCh[n]

This command sets both the Start and End frequency of the notch at the specified index.

NOTE. *An existing Notch and associated values will be overwritten without warning.*

Conditions

This command modifies an existing notch; it does not create the notch. A notch must exist at the given index.

The active plug-in must be Multitone.

Group

Tones

Syntax

```
MTONe:TONes:NOTCh[n] <start>,<end>
MTONe:TONes:NOTCh[n]?
```

Related Commands

WPLugin:ACTive
MTONe:TONes:NOTCh:ADD

Arguments

<start> ::= <NRf>

<end> ::= <NRf>

[n] is an index in the notch table <NR1> (If omitted, interpreted as 1.)

Returns

<start> ::= <NR3>

<end> ::= <NR3>

[n] is an index in the notch table <NR1>

Examples

`MTONE:TONES:NOTCh3 1.5E6,2.5E6` sets the notch at index 3 to a Start frequency of 1.5 MHz and an End frequency of 2.5 MHz.

`MTONE:TONES:NOTCh6?` might return `2.5000000000E+6,2.7000000000E+6` indicating that the notch at index 6 has a Start frequency of 2.5 MHz and an End frequency of 2.7 MHz.

MTONe:TONes:NOTCh[n]:STARt

This command sets or returns the Start frequency of the notch at the specified index.

NOTE. *An existing Notch and associated values will be overwritten without warning.*

Conditions This command modifies an existing notch; it does not create the notch.

The active plug-in must be Multitone.

Group Tones

Syntax

```
MTONe:TONes:NOTCh[n]:STARt <start>
MTONe:TONes:NOTCh[n]:STARt?
```

Related Commands WPLugin:ACTive
MTONe:TONes:NOTCh:ADD

Arguments <start> ::= <NRf>

[n] is an index in the notch table <NR1> (If omitted, interpreted as 1.)

Returns <start> ::= <NR3>

Examples `MTONE:TONES:NOTCH4:START 1.5E9` sets the notch at index 4 to a Start frequency of 1.5 GHz.

`MTONE:TONES:NOTCH3:START?` might return `1.5000000000E+9` indicating the Start frequency for notch 3 is set to 1.5 GHz.

MTONe:TONes:NOTCh[n]:END

This command sets or returns the End frequency of the notch at the specified index.

NOTE. *An existing Notch and associated values will be overwritten without warning.*

Conditions This command modifies an existing notch; it does not create the notch.

The active plug-in must be Multitone.

Group Tones

Syntax

```
MTONe:TONes:NOTCh[n]:END <end>
MTONe:TONes:NOTCh[n]:END?
```

Related Commands WPLugin:ACTive
MTONe:TONes:NOTCh:ADD

Arguments <end> ::= <NRf>

[n] is an index in the notch table <NR1> (If omitted, interpreted as 1.)

Returns <end> ::= <NR3>

Examples `MTONE:TONES:NOTCH3:END 2.8E9` sets the notch at index 4 to an End frequency of 2.8 GHz.

`MTONE:TONES:NOTCH3:END?` might return `6.7000000000E+9` indicating the End frequency for notch 3 is set to 6.7 GHz.

MTONe:COMPile:NAMe

This command sets the name of the signal that is to be compiled.

NOTE. *Duplicate signal names will be overwritten without warning upon compile.*

Conditions The active plug-in must be Multitone.

Group Compile

Syntax

```
MTONe:COMPile:NAMe <signal_name>
MTONe:COMPile:NAMe?
```

Related Commands WPLugin:ACTive

Arguments <signal_name>::=<string>

Returns <signal_name>

Examples `MTONE:COMPILE:NAME "songbird"` loads the name songbird into the name field of the Compile Settings.

`MTONE:COMPILE:NAME?` might return `"tonewfm"` indicating this is the name defined for the complied waveform.

MTONe:COMPile:PLAY

This command enables the “Play after assign” setting in the Compile Settings. When enabled, upon a successful compilation, the instrument will play the signal on the specified channel.

The query form returns the current status.

Conditions When operating from SourceXpress, you must be connected to a real instrument to play a waveform.

The active plug-in must be Multitone.

Group Compile

Syntax `MTONe:COMPile:PLAY {0|1|OFF|ON}`

Related Commands WPLugin:ACTive
MTONe:COMPile:CHANnel

Arguments 0 or OFF disables the play after assign setting.

1 or ON enables the play after assign setting.

Returns A single <Boolean> value representing the setting {0|1}.

Examples `MTONE:COMPILE:PLAY ON` sets the waveform to start playout on the designated channel when compile completes.

`MTONE:COMPILE:PLAY?` might return 0, indicating the waveform is not set to start playing after compiling.

MTONe:COMPile:CASSign

This command sets or returns the state (enabled or disabled) to compile the waveform and immediately assign it to a specified channel (enabled) or just compile the waveform (disabled).

Group Compile

Syntax

```
MTONe:COMPile:CASSign {0|1|OFF|ON}
MTONe:COMPile:CASSign?
```

Related Commands MTONe:COMPile:CHANnel

Arguments

0 or OFF will only compile the waveform.
1 or ON will compile and assign the waveform to a channel.

Returns A single <Boolean> value.

Examples

`MTONE:COMPILE:CASSIGN 1` enables the compile and assign function.

`MTONE:COMPILE:CASSIGN?` might return `0`, indicating that the compile and assign function is disabled.

MTONe:COMPile:CHANnel

This command sets or returns which channel the signal will be assigned to upon compile.

Conditions The active plug-in must be Multitone.

Group Compile

Syntax

```
MTONe:COMPile:CHANnel <Channel>
MTONe:COMPile:CHANnel?
```

Related Commands WPLugin:ACTive

Arguments <channel>::=<NR1> is a valid channel number.

Returns A single <NR1> value.

Examples `MTONE:COMPILE:CHANNEL 1` enables the “Compile and assign to” setting in the Compile Settings and assigns Channel 1.

`MTONE:COMPILE:CHANNEL?` might return `1`, indicating thatwaveforms are assigned to channel 1.

MTONe:COMPile:SRATe

This command sets or returns the sampling rate of the signal in Hz.

Conditions

The active plug-in must be Multitone.

The Sampling Rate for Compile must have Auto Calculate disabled to use the rate set with this command.

Group

Compile

Syntax

```
MTONe:COMPile:SRATe <sample_rate>
MTONe:COMPile:SRATe?
```

Related Commands

WPLugin:ACTive
MTONe:COMPile:SRATe:AUTO

Arguments

<sample_rate>::=<NRf>

The maximum sampling rate is dependent on the instrument and instrument options.

Returns

<sample_rate>::=<NR3>

Examples

`MTONE:COMPILE:SRATE 1E9` sets the sampling rate to 1 GHz.

`MTONE:COMPILE:SRATE?` might return `2.2500000000E+9` indicating the sampling rate is set to 2.25 GHz

MTONe:COMPile:SRATe:AUTO

This command sets or returns if the sampling rate will be automatically calculated at compile time.

Conditions The active plug-in must be Multitone.

Group Compile

Syntax

```
MTONe:COMPile:SRATe:AUTO {0|1|OFF|ON}
MTONe:COMPile:SRATe:AUTO?
```

Related Commands WPLugin:ACTive
MTONe:COMPile:SRATe

Arguments 0 or OFF disables Auto Calculate and uses the Manual sampling rate setting

1 or ON enables Auto Calculate sampling rate setting

Returns A single <Boolean> value representing the setting {0|1}.

Examples `MTONE:COMPILE:SRATE:AUTO ON` enables the Auto Calculate, setting the signal's sampling rate automatically during compile.

`MTONE:COMPILE:SRATE:AUTO?` might return `0`, indicating that the sampling rate will not be automatically set during compile.

MTONe:COMPile:CORRection:APPLy

This command sets or returns if a correction file is to be applied during compile.

Conditions

The active plug-in must be Multitone.

A correction plug-in must be installed.

Group

Compile

Syntax

```
MTONe:COMPile:CORRection:APPLy {0|1|OFF|ON}
MTONe:COMPile:CORRection:APPLy?
```

Related Commands

WPLugin:ACTive
MTONe:COMPile:CORRection:PATH

Arguments

0 or OFF disables the correction file; no corrections will be applied.

1 or ON enables correction file setting, applying corrections during compile.

Returns

A single <Boolean> value representing the setting {0|1}.

Examples

`MTONE:COMPILE:CORRECTION:APPLY ON` will apply a correction file at during compile.

`MTONE:COMPILE:CORRECTION:APPLY?` might return `0`, indicating that no corrections will be applied during compile.

MTONe:COMPile:CORRection:PATH

This command sets or returns the correction file's filepath and filename to use during compile if the Apply Corrections File is enabled.

Conditions The active plug-in must be Multitone.

A correction plug-in must be installed.

Group Compile

Syntax

```
MTONe:COMPile:CORRection:PATH <filepath>
MTONe:COMPile:CORRection:PATH?
```

Related Commands WPLugin:ACTive
MTONe:COMPile:CORRection:APPLy

Arguments <filepath>:=<string>

Returns <string>

Examples `MTONE:COMPILE:CORRECTION:PATH "C:\temp\CorrectionFile.cor"` will set the Correction File's filepath and filename to "C:\temp\CorrectionFile.cor" for use at compile time if the user has chosen to Apply Corrections.

`MTONE:COMPILE:CORRECTION:PATH?` might return `"C:\temp\CorrectionFile.cor"`, indicating the set file and path.

MTONe:COMPile (No Query Form)

This command compiles and generates a waveform using the Multitone plug-in compile settings. The type of waveform compiled (Tones or Chirp) is dependent on the Multitone type setting.

This command enables the **Overwrite existing waveform** setting in the Compile settings.

Conditions The active plug-in must be Multitone.

This is an overlapping command. Overlapping commands run concurrently with other commands, allowing additional commands to start before the overlapping command has finished.

Group Compile

Syntax `MTONe:COMPile`

Related Commands WPLugin:ACTive
MTONe:TYPE

Examples `MTONE:COMPILE` will compile and generate a waveform based on the active type.

MTONe:COMPile:CANCel (No Query Form)

This command cancels a compilation currently in progress.

Conditions The active plug-in must be Multitone.

Group Compile

Syntax `MTONe:COMPile:CANCel`

Related Commands WPLugin:ACTive

Examples `MTONE:COMPILE:CANCEL` immediately ends the current compile process.

Status and Events

Status and Events

There is no status and event information.

Error messages and codes

The following table lists error codes and messages that are unique to the Multitone plug-in.

Error code	Error message
7400	Notch Error. Maximum number of notches. The maximum number of notches has been reached.
7401	Notch Error. Unable to delete, invalid index.
7402	Notch Error. Unable to delete, notch does not exist.
7410	Compile Error. Internal error during compilation of signal.
7411	Compile Error. Number of samples required to create the signal is more than instrument supported maximum length.
7412	Compile Error. Number of samples required to create the signal is less than instrument supported minimum length.
7413	Compile Error. Waveform length required to create the signal does not meet the granularity of instrument waveform length.
7414	Compile Error. Sampling rate required to create the signal exceeds the maximum sampling rate supported by the instrument.
7415	Compile Error. Sampling rate required to create the signal is less than the minimum supported by the instrument.
7416	Compile Error. The sampling rate is not sufficient to create the signal. Increase the sampling rate.

Index