

# General Commands Reference Guide O



# General Commands Reference Guide O

---

TRACE32 Online Help

TRACE32 Directory

TRACE32 Index

TRACE32 Documents .....	
General Commands .....	
General Commands Reference Guide O .....	1
History .....	5
OCP .....	6
OCP	OpenCoreProtocol WatchPoint 6
ON .....	7
ON	Event-controlled PRACTICE script execution 7
Onchip .....	12
Onchip	Trace method Onchip, recording, and analysis commands 12
Onchip-specific Trace Commands .....	13
Onchip.ATTACH	Attach to the onchip trace 13
Onchip.Mode	Set the trace operation mode 13
Onchip Trace Commands .....	17
Onchip.ACCESS	Define access path to program code for trace decoding 17
Onchip.Arm	Arm the trace 17
Onchip.AutoArm	Arm automatically 17
Onchip.AutoInit	Automatic initialization 17
Onchip.BookMark	Set a bookmark in trace listing 17
Onchip.Chart	Display trace contents graphically 17
Onchip.CLOCK	Clock to calculate time out of cycle count information 18
Onchip.ComPare	Compare trace contents 18
Onchip.ComPareCODE	Compare trace with memory 18
Onchip.DISable	Disable the trace 18
Onchip.DisConfig	Trace disassembler configuration 18
Onchip.DRAW	Plot trace data against time 18
Onchip.EXPORT	Export trace data for processing in other applications 18
Onchip.FILE	Load a file into the file trace buffer 19
Onchip.Find	Find specified entry in trace 19
Onchip.FindAll	Find all specified entries in trace 19
Onchip.FindChange	Search for changes in trace flow 19
Onchip.FindProgram	Advanced trace search 19
Onchip.FindReProgram	Activate advanced existing trace search program 19

Onchip.FindViewProgram	State of advanced trace search programming	19
Onchip.FLOWPROCESS	Process flowtrace	20
Onchip.FLOWSTART	Restart flowtrace processing	20
Onchip.GOTO	Move cursor to specified trace record	20
Onchip.Init	Initialize trace	20
Onchip.JOINFILE	Concatenate several trace recordings	20
Onchip.List	List trace contents	20
Onchip.ListNesting	Analyze function nesting	20
Onchip.ListVar	List variable recorded to trace	21
Onchip.LOAD	Load trace file for offline processing	21
Onchip.MERGEFILE	Combine two trace files into one	21
Onchip.OFF	Switch off	21
Onchip.PlatformCLOCK	Set clock for platform traces	21
Onchip.PROfileChart	Profile charts	21
Onchip.PROfileSTATistic	Statistical analysis in a table versus time	21
Onchip.PROTOcol	Protocol analysis	22
Onchip.PROTOcol.Chart	Graphic display for user-defined protocol	22
Onchip.PROTOcol.Draw	Graphic display for user-defined protocol	22
Onchip.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	22
Onchip.PROTOcol.Find	Find in trace buffer for user-defined protocol	22
Onchip.PROTOcol.List	Display trace buffer for user-defined protocol	22
Onchip.PROTOcol.PROfileChart	Profile chart for user-defined protocol	22
Onchip.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	23
Onchip.PROTOcol.STATistic	Display statistics for user-defined protocol	23
Onchip.REF	Set reference point for time measurement	23
Onchip.RESet	Reset command	23
Onchip.SAVE	Save trace for postprocessing in TRACE32	23
Onchip.SelfArm	Automatic restart of trace recording	23
Onchip.SIZE	Define buffer size	23
Onchip.SnapShot	Restart trace capturing once	24
Onchip.state	Display trace configuration window	24
Onchip.STATistic	Statistic analysis	24
Onchip.STREAMCompression	Select compression mode for streaming	24
Onchip.STREAMFILE	Specify temporary streaming file path	24
Onchip.STREAMFileLimit	Set size limit for streaming file	24
Onchip.STREAMLOAD	Load streaming file from disk	24
Onchip.STREAMSAVE	Save streaming file to disk	25
Onchip.TDelay	Trigger delay	25
Onchip.TestFocus	Test trace port recording	25
Onchip.TestUtilization	Tests trace port utilization	25
Onchip.Timing	Waveform of trace buffer	25
Onchip.TraceCONNECT	Select on-chip peripheral sink	25
Onchip.TRACK	Set tracking record	25

Onchip.TRIGGER	Trigger the trace	26
Onchip.View	Display single record	26
Onchip.ZERO	Align timestamps of trace and timing analyzers	26
<b>Onchip2</b> .....		<b>27</b>
Onchip2	Second onchip trace buffer	27

## History

---

- 20-Dec-21 Description of [Onchip.Mode STREAM](#) and [Onchip.Mode LeashSTREAM](#) was added.
- 16-Dec-19 Description for command [Onchip.ATTACH](#) added.

For a description of the **OCP** commands, see [“System Trace User’s Guide”](#) (trace\_stm.pdf).

```

Format:          ON <event> [<action>]

<event>:        <device_specific_events>
                 <practice_specific_events>
                 <cpu_specific_events>

<device_
specific_
events>:        ABREAK
                 OBREAK
                 CORESWITCH
                 GO
                 PBREAK
                 PBREAKAT <address>
                 POWERDOWN
                 POWERUP
                 RESET
                 SYSDOWN
                 SYSUP
                 TRIGGER
                 ATRIGGER
                 OTRIGGER
                 CATRIGGER

<action>:       inherit
                 CONTinue
                 DO <file>
                 GOSUB <label> | <block>
                 GOTO  <label> | <block>
                 JUMPTO <label> | <block>
                 DEFault

```

The **ON** command enables the automatic start or branching of the PRACTICE scripts controlled by several events. The registered actions are stored on the PRACTICE stack, therefore the command is only valid in the block in which it was set, and in the subroutines called in this block. The currently active **ON** command can be viewed with the **PMACRO** command. The PRACTICE script will be started and stopped automatically, if the **GOSUB** action is used. If no target label is given, the line or block after the **ON** command will be executed instead.

**NOTE:**

If you want the action to remain permanently active, use the **GLOBALON** command.

Using the **GLOBALON** command, you can create actions for global events, which are available for an entire TRACE32 session.

**Events: <device\_specific\_events>**


---

Device-specific Events	Descriptions
<b>ABREAK</b>	The analyzer state changed to the break state.
<b>ATRIGGER</b>	Responds to an analyzer trigger.
<b>CATRIGGER</b>	Responds to an CAnalyzer trigger.
<b>CORESWITCH</b>	SMP debugging: The currently displayed context changed to a different core or thread.
<b>GO</b>	The target program started.
<b>OBREAK</b>	The on-chip state changed to the break state.
<b>OTRIGGER</b>	Responds to an on-chip trigger.
<b>PBREAK</b>	The target program stopped.
<b>PBREAKAT</b>	The target program stopped at a specific address.
<b>POWERDOWN</b>	Target power is switched off.
<b>POWERUP</b>	Target power is switched on.
<b>RESET</b>	A target reset was detected.
<b>SYSDOWN</b>	System mode changed to <b>Down</b> or <b>NoDebug</b> . The event is also triggered if the debugger is in system mode <b>StandBy</b> and the target power is switched off.
<b>SYSUP</b>	System mode changed to <b>Up</b> . The event is also triggered if the debugger is in system mode <b>StandBy</b> and the target power is switched on.
<b>TRIGGER</b>	A podbus trigger occurred (internal or external source can be selected via TRIGGER window).



## Events: <practice\_specific\_events>

---

<practice_specific_events>	For a description of the PRACTICE specific events, such as <b>ON ERROR</b> , refer to <b>ON</b> in the “ <b>PRACTICE Script Language Reference Guide</b> ” (practice_ref.pdf).
----------------------------	--

## Events: <cpu\_specific\_events>

---

<cpu_specific_events>	For information about CPU specific events, refer to the <b>Processor Architecture Manual(s)</b> <a href="#">[▲]</a> listed in the <b>See also</b> block below.
-----------------------	--

## <action>

---

One of the following actions can be defined for any of the above events:

Actions	Descriptions
<b>inherit</b>	An already defined action for this event in the current stack level will be removed. If an action is registered in a higher stack level, the action of the higher stack level will be inherited. If no action is registered in any higher stack level, the debugger's default action will be performed (e.g. stop on error event).
<b>CONTinue</b>	If the event occurs, script execution will be continued. Use this option e.g. to ignore errors which would cause the script execution to halt.
<b>DO</b>	If the event occurs, the specified PRACTICE script file will be executed automatically.
<b>GOSUB</b>	If the event occurs, a subroutine call will occur. The subroutine can be specified as label, or inline as PRACTICE block. With RETURN, the subroutine will return normal program execution. For the ERROR event, the subroutine will return to the line after the command which caused the error.
<b>GOTO</b>	If the event occurs, the script execution will continue at the specified label, or in the specified inline PRACTICE block.
<b>JUMPTO</b>	If the event occurs, the script execution will continue at the specified label, or in the specified inline PRACTICE block. Subroutine calls and block nestings are removed from the PRACTICE stack. Use this action e.g. as global error/exception handler.
<b>DEFault</b>	The debugger's default action will be performed if the specified event occurs. Actions defined in a higher stack level for this event will be ignored. Use this action e.g. inside an event handler subroutine to avoid re-entry while the handler's subroutine is active.

## Examples

### Example 1: Automatically set up debugger and target after a power-up:

```
ON POWERUP GOSUB
(
  SYStem.Up
  Data.LOAD.ELF project.x           ;load project and
  Register.Set PC main              ;set up registers
  Register.Set SP 0x100
  Data.Set 0x00FFFA21 0x40         ;disable watchdog
  WINCLEAR
  Data.List
  RETURN
)
STOP
```

### Example 2: Manipulate program execution:

```
ON PBREAKAT ADDRESS.OFFSET(sieve) GOTO
(
  Var.set NumCalls++
  Var.set %SPaces isascii = (name[0]>=0x20 && name[0]<='z') ? 1 : 0;
  Go
)
STOP
```

### Example 3: React on trigger event:

```
Trigger.Connect In                ;configure trigger connector as input
Trigger.Mode Falling

ON TRIGGER GOSUB
(
  ON TRIGGER DEFault              ;disable action while subroutine active
  if SYStem.Up()&&STATE.RUN()
  (
    Break
    Register.Set PC  Data.Long(SD:0x40000204)
    Register.Set MSR Data.Long(SD:0x40000208)
  )
  RETURN
)
STOP
```

#### Example 4: Perform action when core stops:

```
ON PBREAK GOTO
(
    PRINT "Stopped at address " Register(PC)
    STOP
)
STOP
```

#### Example 5: Close output of terminal window to file at next program stop:

```
TERM.METHOD SingleE 0x3ffd 0x3ffe
TERM.SIZE 120. 3000.
TERM.view
TERM.WRITE 0x3ffd out.txt
ON PBREAK GOTO TERMCLOSE
Go
STOP
TERMCLOSE: TERM.CLOSE
...
```

#### See also

---

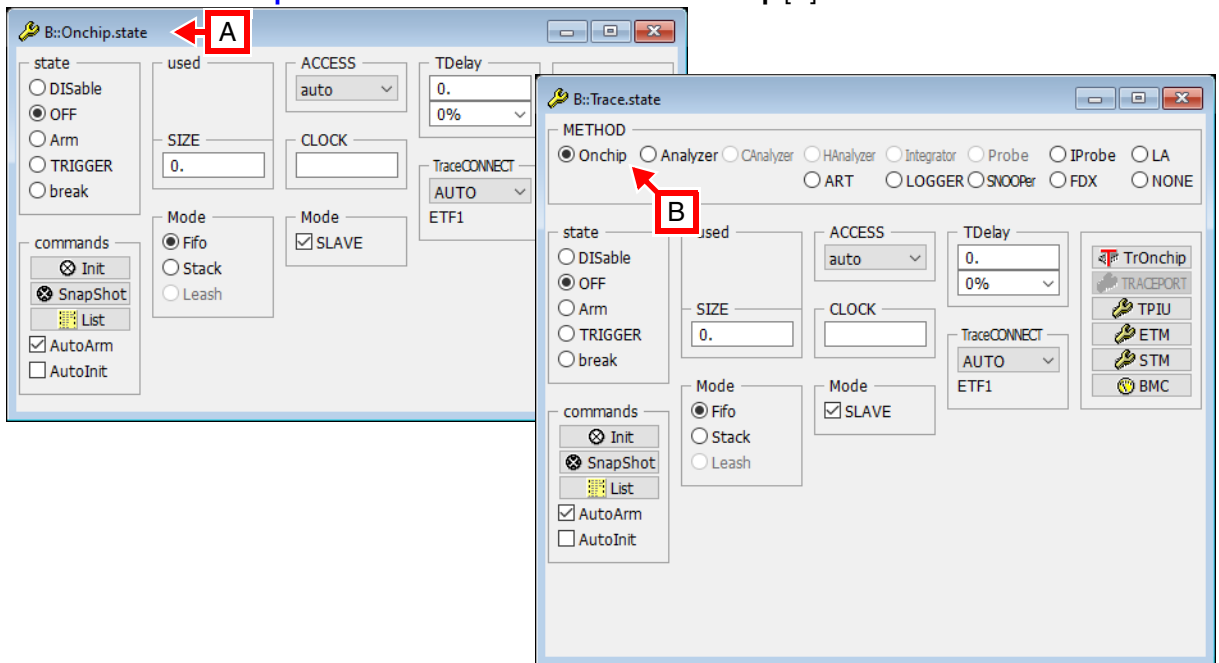
##### ■ GLOBALON

- ▲ 'Mico32 specific Event for the ON and GLOBALON Command' in 'Mico32 Debugger'
- ▲ 'CPU specific Events for the ON and GLOBALON Command' in 'Intel® x86/x64 Debugger'

Some processors are equipped with an on-chip trace memory. The **Onchip** trace method can be used in this case to read, analyze and display the trace information. In the rare case the target processor provides more than one on-chip trace buffers, the following ones can be controlled using the **Onchip2**, **Onchip3** ... command groups.

For selecting and configuring the trace method Onchip, use the TRACE32 command line or a PRACTICE script (\*.cmm) or the **Onchip.state** window [A].

Alternatively, use the **Trace.state** window: click the option **Onchip** or execute the command **Trace.METHOD Onchip** in order to select the trace method **Onchip** [B].



The chapter "**Onchip-specific Trace Commands**", page 13 describes the Onchip-specific configuration commands. While the chapter "**Onchip Trace Commands**", page 17 lists the Onchip trace analysis and display commands, which are shared with other TRACE32 trace methods.

### See also

- [Trace.METHOD](#)
- ▲ ['Onchip Trace Commands' in 'General Commands Reference Guide O'](#)
- ▲ ['Release Information' in 'Release History'](#)

# Onchip-specific Trace Commands

---

## Onchip.ATTACH

Attach to the onchip trace

---

Format:           **Onchip.ATTACH**

This command should be used before a **SYStem.Mode Attach** if the Onchip trace has already been set up by the target application and should not be reset by the debugger.

The command **Onchip.ATTACH** can only be executed from the **DISable** state (**Onchip.DISable**).

## Onchip.Mode

Set the trace operation mode

---

Format:           **Onchip.Mode** [*<mode>*]

*<mode>*:           **Fifo**  
                  **Stack**  
                  **Leash**  
                  **STREAM**  
                  **LeashSTREAM**  
                  **SLAVE [ON | OFF]**  
  
                  *<specific\_modes>*

Selects the trace operation mode.

- |              |  |
|--------------|--|
| <b>Fifo</b>  | If the trace is full, new records will overwrite older records. The trace records always the last cycles before the break. |
| <b>Stack</b> | If the trace is full recording will be stopped. The trace always records the first cycles after starting the trace.        |
| <b>Leash</b> | Stops the program execution when trace is nearly full.   |

## STREAM

Streams the trace data to the streaming file on the host while recording.

STREAM mode can only be used:

- If the onchip trace memory can be read while the program execution is running.
- If the trace memory fills up slower than it can be streamed. This usually requires the trace information to be filtered.

The streaming file is placed into the TRACE32 temp directory (**OS.PresentTemporaryDirectory()**) by default and is named `<trace32_instance_id>streamo.t32 (OS.ID())`. If you explicitly want to specify a location for the streaming file use the command `<trace>.STREAMFILE <file>`.

The command is only enabled, if this operation mode is supported by the core architecture used.

## LeashSTREAM

LeashSTREAM allows to accumulate a number of onchip trace recordings to analyze them as a whole.

LeashSTREAM combines the two operation modes Leash and STREAM. When activated, TRACE32 starts the program execution in **StopAndGo** mode. Whenever the onchip trace memory is almost full, the program is stopped briefly and the current trace data is added to the streaming file. Please be aware that recording time will usually be much shorter than streaming time.

The command is only enabled, if this operation mode is supported by the core architecture used.

## SLAVE OFF

Separates the trace from the program execution, i.e. trace is recording even when the program execution is stopped (very rarely used command).

## SLAVE ON

Ties the trace to the execution of the program, i.e. trace and the filter/trigger work only during user program execution (very rarely used command).

*<specific\_modes>*

Processor architecture specific modes. Please refer to your **Processor Architecture Manual** for more information.

## Example for Trace.Mode LeashSTREAM

```
SYStem.CPU R7F701035 ; core architecture used is RH850
...
SYStem.Option PERSTOP ON ; suspends the peripherals when the
; program execution is stopped;
; this is useful to prevent timer
; exceptions

Trace.Method.Onchip

NEXUS.SUSpend ON ; stalls the program execution
; whenever the on-chip NEXUS-FIFO
; threatens to overflow

; configure chip time stamps

Trace.CLOCK 100.MHz ; TRACE32 can calculate times
; out of Nexus time stamps;
; TRACE32 has to be informed about
; the appropriate clock frequency

NEXUS.TimeStamps.ON ; add time stamps to Nexus messages

Trace.Mode.LeashSTREAM

Go

... ; run program for some time

Break

...

COVerge.ADD ; add accumulated trace information
; to TRACE32 code coverage system

...

Trace.Chart.sYmbol ; display time chart of accumulated
; trace information
```

record	run	address	cycle	data	symbol	ti.back
+00000000028243		P:FEDF8ACE	ptrace		\\sieve\sieve\sieve+0x4	0.080us
		mov		#-0x1204854,r5		
		add		r2,r5		
		mov		#0x1,r6		
		st.b		r6,0x0[r5]		
		add		#0x1,r2		
		addi		#-0x13,r2,r0		
		blt		0xFEDF8ACE		
+00000000028244		P:FEDF8ACE	ptrace		\\sieve\sieve\sieve+0x4	0.190us
		mov		#-0x1204854,r5		
		add		r2,r5		
+00000000028245	BRK					
+00000000028248		P:FEDF8AD6	ptrace		\\sieve\sieve\sieve+0x0C	0.100us
		mov		#0x1,r6		
		st.b		r6,0x0[r5]		
		add		#0x1,r2		
		addi		#-0x13,r2,r0		
		blt		0xFEDF8ACE		
+00000000028250		P:FEDF8ACE	ptrace		\\sieve\sieve\sieve+0x4	0.170us

The streaming time is marked with BRK in the **Trace.List** representation.

```
Trace.FindAll , TRACEBRK
```

### See also

- [<trace>.Mode](#)
- [<trace>.STREAMFILE](#)
- ▲ ['Onchip Trace' in 'APS Debugger'](#)
- ▲ ['Cortex-M specific Onchip Commands' in 'Cortex-M Debugger'](#)
- ▲ ['Trace' in 'SH2, SH3 and SH4 Debugger'](#)
- ▲ ['On-chip Trace' in 'StarCore Debugger and Trace'](#)



## **Onchip.ACCESS** Define access path to program code for trace decoding

---

See command [<trace>.ACCESS](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 130).

## **Onchip.Arm** Arm the trace

---

See command [<trace>.Arm](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 132).

## **Onchip.AutoArm** Arm automatically

---

See command [<trace>.AutoArm](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 133).

## **Onchip.AutoInit** Automatic initialization

---

See command [<trace>.AutoInit](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 138).

## **Onchip.BookMark** Set a bookmark in trace listing

---

See command [<trace>.BookMark](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 139).

## **Onchip.Chart** Display trace contents graphically

---

See command [<trace>.Chart](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 142).

## **Onchip.CLOCK** Clock to calculate time out of cycle count information

---

See command [<trace>.CLOCK](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 189).

## **Onchip.ComPare** Compare trace contents

---

See command [<trace>.ComPare](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 190).

## **Onchip.ComPareCODE** Compare trace with memory

---

See command [<trace>.ComPareCODE](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 192).

## **Onchip.DISable** Disable the trace

---

See command [<trace>.DISable](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 195).

## **Onchip.DisConfig** Trace disassembler configuration

---

See command [<trace>.DisConfig](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 196).

## **Onchip.DRAW** Plot trace data against time

---

See command [<trace>.DRAW](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 199).

## **Onchip.EXPORT** Export trace data for processing in other applications

---

See command [<trace>.EXPORT](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 210).

See command [<trace>.FILE](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 229).

See command [<trace>.Find](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 231).

See command [<trace>.FindAll](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 233).

See command [<trace>.FindChange](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 234).

See command [<trace>.FindProgram](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 235).

See command [<trace>.FindReProgram](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 235).

See command [<trace>.FindViewProgram](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 236).

See command [<trace>.FLOWPROCESS](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 237).

**Onchip.FLOWSTART**Restart flowtrace processing

---

See command [<trace>.FLOWSTART](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 237).

**Onchip.GOTO**Move cursor to specified trace record

---

See command [<trace>.GOTO](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 240).

**Onchip.Init**Initialize trace

---

See command [<trace>.Init](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 242).

**Onchip.JOINFILE**Concatenate several trace recordings

---

See command [<trace>.JOINFILE](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 242).

**Onchip.List**List trace contents

---

See command [<trace>.List](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 244).

**Onchip.ListNesting**Analyze function nesting

---

See command [<trace>.ListNesting](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 257).

## Onchip.ListVar

List variable recorded to trace

---

See command [<trace>.ListVar](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 260).

## Onchip.LOAD

Load trace file for offline processing

---

See command [<trace>.LOAD](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 264).

## Onchip.MERGEFILE

Combine two trace files into one

---

See command [<trace>.MERGEFILE](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 266).

## Onchip.OFF

Switch off

---

See command [<trace>.OFF](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 271).

## Onchip.PlatformCLOCK

Set clock for platform traces

---

See command [<trace>.PlatformCLOCK](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 273).

## Onchip.PROfileChart

Profile charts

---

See command [<trace>.PROfileChart](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 277).

## Onchip.PROfileSTATistic

Statistical analysis in a table versus time

---

See command [<trace>.PROfileSTATistic](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 316).

See command [<trace>.PROTOcol](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 333).

---

**Onchip.PROTOcol.Chart**

Graphic display for user-defined protocol

See command [<trace>.PROTOcol.Chart](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 333).

---

**Onchip.PROTOcol.Draw**

Graphic display for user-defined protocol

See command [<trace>.PROTOcol.Draw](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 335).

---

**Onchip.PROTOcol.EXPORT**

Export trace buffer for user-defined protocol

See command [<trace>.PROTOcol.EXPORT](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 336).

---

**Onchip.PROTOcol.Find**

Find in trace buffer for user-defined protocol

See command [<trace>.PROTOcol.Find](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 337).

---

**Onchip.PROTOcol.List**

Display trace buffer for user-defined protocol

See command [<trace>.PROTOcol.List](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 338).

---

**Onchip.PROTOcol.PROfileChart**

Profile chart for user-defined protocol

See command [<trace>.PROTOcol.PROfileChart](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 341).

## **Onchip.PROTOcol.PROfileSTATistic** Profile chart for user-defined protocol

---

See command [<trace>.PROTOcol.PROfileSTATistic](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 342).

## **Onchip.PROTOcol.STATistic** Display statistics for user-defined protocol

---

See command [<trace>.PROTOcol.STATistic](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 344).

## **Onchip.REF** Set reference point for time measurement

---

See command [<trace>.REF](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 351).

## **Onchip.RESet** Reset command

---

See command [<trace>.RESet](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 351).

## **Onchip.SAVE** Save trace for postprocessing in TRACE32

---

See command [<trace>.SAVE](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 352).

## **Onchip.SelfArm** Automatic restart of trace recording

---

See command [<trace>.SelfArm](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 356).

## **Onchip.SIZE** Define buffer size

---

See command [<trace>.SIZE](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 366).

See command [<trace>.SnapShot](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 366).

## Onchip.state

## Display trace configuration window

---

See command [<trace>.state](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 369).

## Onchip.STATistic

## Statistic analysis

---

See command [<trace>.STATistic](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 371).

## Onchip.STREAMCompression

## Select compression mode for streaming

---

See command [<trace>.STREAMCompression](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 475).

## Onchip.STREAMFILE

## Specify temporary streaming file path

---

See command [<trace>.STREAMFILE](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 476).

## Onchip.STREAMFileLimit

## Set size limit for streaming file

---

See command [<trace>.STREAMFileLimit](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 477).

## Onchip.STREAMLOAD

## Load streaming file from disk

---

See command [<trace>.STREAMLOAD](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 478).



See command [<trace>.STREAMSAVE](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 480).

**Onchip.TDelay**Trigger delay

---

See command [<trace>.TDelay](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 481).

**Onchip.TestFocus**Test trace port recording

---

See command [<trace>.TestFocus](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 484).

**Onchip.TestUtilization**Tests trace port utilization

---

See command [<trace>.TestUtilization](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 487).

**Onchip.Timing**Waveform of trace buffer

---

See command [<trace>.Timing](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 489).

**Onchip.TraceCONNECT**Select on-chip peripheral sink

---

See command [<trace>.TraceCONNECT](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 491).

**Onchip.TRACK**Set tracking record

---

See command [<trace>.TRACK](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 492).

See command [<trace>.TRIGGER](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 492).

See command [<trace>.View](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 494).

See command [<trace>.ZERO](#) in 'General Commands Reference Guide T' (general\_ref\_t.pdf, page 495).

The **Onchip2** command group is used when the chip provides two onchip trace buffers in order to control the second trace buffer Please refer to [Onchip](#) for information about the available sub-commands.