xPC Target Quick Reference Guide for v2.5 (R14.0)

Main Libraries : **xpclib, xpctgboxlib** (*.mdl)

Main Example : **xpcosc** (*.mdl) *(see >> help xpcdemos for more)*

➢ parsweepdemo	(*.m)	- Demonstrates parameter updates in xPC
➢ dataloggingdemo	(*.m)	- Demo time- and value-equidistant data logging
➢ scfreerundemo	(*.m)	- Demo FreeRun display mode of xPC Scope
➢ scprepostdemo	(*.m)	- Demo pre/post triggering of xPC scope
➢ scscopedemo	(*.m)	- Demo scope triggered xPC Scope
➢ scsignaldemo	(*.m)	- Demo signal triggered xPC scope
➢ scsoftwaredemo	(*.m)	- Demo software triggered xPC Scope
➢ tgscopedemo	(*.m)	- Demo of xPC TargetScope
➢ xpcbench	(*.m)	- Execute xPC Target benchmarks and show result
➢ xpcdngdemo	(*.m)	- demo Dials & Gauges model with xPC Target
➢ xpcspectrumdemo	(*.m)	- Spectrum Analyzer demo with xPC Target

Getting Help about (.m)-files >>help M-fileNAME or >>doc M-fileNAME*

- ➢ **xpc_osc1-3**.mdl (Tutorial)
- ➢ **xpctank**.mdl
- ➢ **xpctankpanel**.mdl (D&G)
- ➢ ***xpccan*pc104**.mdl (CAN)
- ➢ ***xpccan*isa**.mdl (")
- ➢ ***xpccan*pci**.mdl (")
- ➢ ***xpccan*pcififo1-6**.mdl ("-PCI)
- ➢ ***xpccan*104fifo1-6.mdl** ("-104)
- ➢ ***xpccan*pci1byte**.mdl
- ➢ ***xpccan*intpc104**.mdl
- ➢ ***xpccan*intpci**.mdl

- ➢ **xpcdspspectrum**.mdl
- ➢ **xpcasyncbuffer**.mdl
- ➢ **xpcasynctrans**.mdl
- ➢ **xpcrs232**.mdl
- ➢ **xpcrs232bindemo**.mdl
- ➢ **xpcemptymdl**.mdl (xPC Template)

- ➢ . . . *(complete list see Page 7)*

Overview of important Commands:

• xpcroot	[Path to installed xPC Target]
• findmsvc	[locate installed MC Visual Compiler]
• xpcsetup	{ Target xPC Settings }
• xpcexplr	{ xPC Target Explorer Tool }
• getxpcenv	[gets xPC current Properties]
• setxpcenv	[set xPC Properties, >>updatexpcenv must follow !]
• updatexpcenv	[Update of before set xPC based Target Properties]
• xpcbootdisk	[create command line based a xPC BootDisk]
• xpctargetping	[TCP/IP as well as using RS232]
• xpctest or xpctest('noreboot')	[Full automatic Test with xpcosc.mdl]
• getxpcpci	[installed I/O Boards information through xPC]
• - - - - - - - - - - -	
• xpctargetspy	{ Target Screen Spy }
• xpcrctool	{ xPC Target Remote Control Tool }
• - - - - - - - - - - -	
• xpc**tg**scope	[tg here = Target, not the xpctarget object variable !]

- xpcscope { Host Scope Manager }
- xpcwwwenable { closes MATLAB Target connection }
- - - - - - - - - - - -
- **tgob = xpc;** [creates xPC Target object in ML-Workspace]
- load(tgob,'AppNameAlreadyCompiled=dlm-Filename')
- unload(tgob)

- +tgob ≈ start(trgob) ≈ tgob.start
- -tgob ≈ stop(trgob) ≈ tgob.stop

- tgob.stoptime = 1000 ≈ set(tgob,’stoptime’,1000)
- tgob.sampletime = 0.01 ≈ set(tgob,’sampletime’,0.01)
- **tgob.showsignals = ‘on’** ≈ set(tgob,’showsignals’,’on’)
- **tgob.showparameter= on'** ≈ set(tgob,'showparameter','on')

- sc1 = tgob.addscope(‘target’,1) ≈ sc1 = addscope(tgob,’target’,1)
- +sc1 / -sc1 ≈ sc1.start / sc1.stop ≈ start(sc1) / stop(sc1)
- - - - - - -
- sc1 = getscope(tgob,1) ≈ sc1 = tgob.getscope(1)
- sc1.Mode = 'numerical'

- tgob.addsignal(sc1,[0,1]) ≈ addsignal(sc1,[0,1])
- **setparam(tgob,5, 800)**
- **getparam(tgob,5)**
- tgob.AvgTET ≈ get(tgob,’AvgTET’)
- - - - - - - - - - - -
- xpctargetboxinit [special Settings for xPC TargetBox HW]

- fs_xpc = xpctarget.fs [creates an **F**ile**S**ystem xPC Object]
  - **Methods**:
    bufferinfo cd comm
    dir diskinfo fclose
    fileinfo filetable fopen
    fread fs fwrite
    getfilesize mkdir pwd
    removefile rmdir selectdrive

  - Td = fs_xpc.dir; *for i=1:length(Td); TdNs=Td(i).name; disp(TdNs); end*
  - fs_xpc.pwd
  - fs_xpc.cd('c:\work') ≈ cd(fs_xpc, 'c:\workdir')

- ftp_xpc = xpctarget.ftp [creates an **FTP** FileSystem xPC Object]
  - **Methods**:
    cd dir ftp get mkdir put pwd rmdir

  - ftp_xpc.put('file_from_host_to_target')
    - *put(ftp_xpc,'xpcosc.rtb'); put(ftp_xpc,'autoexec.bat'); put(ftp_xpc,'xpcboot.com');*
    - *put(ftpxpc,'xpctgo1.rtb'); put(ftpxpc,'autoexec.bat'); put(ftpxpc,'checksum.dat'); put(ftpxpc,'xpcboot.com');*
  - ftp_xpc.get('c:\workdir\file_from_target_to_host.dat')

- readxpcfile('tgloggeddata.dat') [interpret the raw data from Target fread method]

- sc1_xpc = xpctarget.xpcsctg(tg,1)
  - **Methods:**

    addsignal
    buildit
    display
    remsignal
    start
    static_display
    static_start
    stop
    trigger
    uminus
    uplus
    xpcsc
    xpcsctg

Note: ***tgob*** user-defind (!) in Simulink → Tools →RTW →Options →In Category: xPC Target code generation Options →Name for xPC Target object created by build process ⇒ enter name or alias here (for these example **tgob** selected) !!

Target Commands

h	Help
c	Command Window Activation
g	Toggle Grid of Scope
v	Toggle Viewmode of S
m	Toggle Scopemode
s	Software Trigger
F#	Select Scope (i.e. F1 → 1st Scope)

Note: Mouse Cursor must be in Scope Area!

Command	**Description**
delallvar	Delete all variables. ***Syntax***: **delallvar**
delvar	Delete a variable. ***Syntax***: **delvar variable_name**
getpar	Displays the value of a block parameter using the parameter index. ***Syntax***: **getpar parameter_index** Example: getpar 24 ≅ P24
getvar	Display the value of a variable. ***Syntax***: **getvar variable_name**
P#	Display or change the value of a block parameter. ***Syntax***: **parameter_name**, or **parameter_name = floating_point_number** Example: P2 or P2 = 1.23e-4
S#	Displays the value of a signal. ***Syntax***: **signal_name**. Example, S2.
sampletime	Enter a new sample time. ***Syntax***: **sampletime = floating_point_number**
setpar	Changes the value of a block parameter using the parameter index. ***Syntax***: **setpar parameter_index = floating_point_number** Example: setpar 24 = a20
setvar	Sets a variable to a value. Later you can use that variable to do a macro expansion. ***Syntax***: **setvar variable_name = target_pc_command** For example, you can type : setvar aa = startscope 2, setvar bb = stopscope 2 or setvar a09 = 9, servar a20 = 20 setvar on = p7= 1 setvar off = p7= 0 => Motor On = type: ON
showvar	Display a list of variables. ***Syntax***: **showvar**
stoptime	Enter a new stop time. Use inf to run the target application until you manually stop it or reset the target PC. ***Syntax***: **stoptime = floating_point_number**
viewmode	Zoom in to one scope, or zoom out to all scopes. ***Syntax***: **viewmode scope_number**, or **viewmode 'all'**

>>doc xpc => xPC Target (im Contents Registerkarte)

- ⇨ „Using xPC Target „
- ⇨ "Advanced Topics"
- ⇨ „Target PC Command-Line Interface" ≅ [ch_adv23.html]

Scope Object Properties (Advanced Topics)

Target PC Command	**MATLAB** Equivalent
`numsamples scope_index = number`	`sc.NumSamples = number`
`decimation scope_index = number`	`sc.Decimation = number`

`scopemode scope_index = 0` or `numerical,`
`1` or `redraw, 2` or `sliding, 3` or `rolling`
	`sc.Mode = 'numerical', 'redraw', 'sliding', 'rolling'`

`triggermode scope_index = 0, freerun,`
`1 software, 2, signal, 3, scope`
	`sc.TriggerMode = 'freerun', 'software', 'signal', 'scope'`

`numprepostsamples scope_index = number`	`sc.NumPrePostSamples = number`
`triggersignal scope_index = signal_index`	`sc.TriggerSignal = signal_index`
`triggersample scope_index = number`	`sc.TriggerSample = number`
`triggerlevel scope_index = number`	`sc.TriggerLevel = number`

`triggerslope scope_index = 0, either,`
`1, rising, 2, falling`
	`sc.TriggerSlope = 'Either', 'Rising', 'Falling'`

`triggerscope scope_index2 = scope_index1`	`sc.TriggerScope = scope_index1`
`triggerscopesample scope_index= integer`	`sc.TriggerSample = integer`
Press function key for scope, and then press **S** or move mouse into the scope window	`sc.trigger`

Scope Object Methods (Advanced Topics)

Target PC Command	**Description and Syntax**	**MATLAB Equivalent**

addscope	`addscope` *`scope_index`*
	`addscope`
		`tg.addscope(scope_index)`
		`tg.addscope`

remscope	`remscope` *`scope_index`*
	`remscope` *`all`*
		`tg.remscope(scope_index)`
		`tg.remscope`

startscope	`startscope` *`scope_index`*	`sc.start` or `+sc`
stopscope	`stopscope` *`scope_index`*	`sc.stop` or `-sc`

addsignal	`addsignal` *`scope_index = signal_index1, signal_index2, ...`*
	`Example:` ***`addsignal 2 = p24`***
		`sc.addsignal(signal_index_vector)`

remsignal	`remsignal` *`scope_index = signal_index1, signal_index2, ...`*	`sc.remsignal(signal_index_vector)`
viewmode	Zoom in to one scope, or zoom out to all scopes. Syntax: `viewmode` *`scope_index`* or left-click the scope window `viewmode` *`'all'`* or right-click any scope window Press function key for scope, and then press **V** to toggle viewmode	

ylimit	`ylimit` *`scope_index`*
	`ylimit` *`scope_index = auto`*
	`ylimit` *`scope_index = num1, num2`*	

grid	`grid` *`scope_index on`*
	`grid` *`scope_index off`*	

```
>> help xpc

% xPC Target          Content of  … xpc/Contents.m
% Version 2.5 (R14) 05-May-2004
%
% xPC Target Environment
%   getxpcenv        - Gets xPC Target Environment Properties.
%   setxpcenv        - Sets xPC Target Environment Properties.
%   updatexpcenv     - Updates the xPC Target Environment.
%   xpcbootdisk      - Creates xPC Target Boot Floppy Disk.
%
% xPC Target Graphical User Interfaces
%   xpcexplr       - xPC Target Explorer.
%   xpcsetup         - GUI to maintain xPC Target Environment.
%   xpcrctool        - xPC Target Remote Control Tool GUI.
%   xpcscope         - xPC Target Host Scope GUI.
%   xpctgscope       - xPC Target Target Scope GUI.
%   xpctargetspy     - Shows the target screen on the host.
%
% xPC Target Object methods
%   xpc              - Construct xPC target object.
%   xpc/get          - Gets value of target object property.
%   xpc/set          - Sets value of target object property.
%   xpc/load         - Loads an application onto the target.
%   unload           - Unloads the current application from the target.
%   xpc/start        - Starts target application execution.
%   xpc/stop         - Stops execution of the target application.
%   addscope         - Adds a scope to the current simulation.
%   getscope         - Gets an xPC scope object.
%   remscope         - Removes a scope from the target.
%   getlog           - Gets part of any of the various simulation logs.
%   getparamid       - Gets the parameter index in the parameter list.
%   getsignalid      - Gets the signal index in the signal list.
%   xpc/close            - Closes the serial port connection to the target.
%   reboot           - Reboots the target system.
%
% xPC Target Scope Object Properties
%   xpcsc/get        - Gets value of scope object property.
%   xpcsc/set        - Sets value of scope object property.
%   xpcsc/start      - Starts xPC Target scope.
%   xpcsc/stop       - Stops xPC Target scope.
%   addsignal        - Adds a (vector of) signal(s) to the scope.
%   remsignal        - Removes signals from scopes.
%   trigger          -  (Software) Triggers one or more xPC scopes.
%
% xPC Target Demos
%   scfreerundemo    - Demonstrates FreeRun display mode of xPC Scope.
%   scscopedemo      - Demonstrates scope triggered xPC Scope.
%   scsignaldemo     - Demonstrates signal triggered xPC scope.
%   scsoftwaredemo   - Demonstrates software triggered xPC scope.
%   scprepostdemo    - Demonstrates pre/post triggering of xPC scope.
%   tgscopedemo      - Demonstration of xPC TargetScope.
%   parsweepdemo     - Demonstrates parameter updates in xPC.
%   dataloggingdemo  - Demonstrates time- and value-equidistant data logging.
%
%   xpcdngdemo       - How to use the demo gauges model with xPC Target
%   xpcspectrumdemo  - How to use the Spectrum Analyzer demo with xPC Target
%   xpcbench         - Execute xPC Target benchmarks and show result
%
% Miscellaneous Functions
%   xpctest          - xPC Target Test Suite.
%   xpctargetping    - 'Ping' the target to test connection.
%   getxpcpci        - Query target PC for installed PCI-boards.
%   xpcwwwenable     - Enables the use of the xPC Target WWW Interface.
%   xpcsliface       - Generates a skeleton Simulink instrumentation Model.
```

```
>> help xpcdemos
```

```
% xPC Target      -- demos and sample script files.
%
% scfreerundemo   - Demonstrates FreeRun display mode of xPC Scope.
% scscopedemo     - Demonstrates scope triggered xPC Scope.
% scsignaldemo    - Demonstrates signal triggered xPC scope.
% scsoftwaredemo  - Demonstrates software triggered xPC scope
% scprepostdemo   - Demonstrates pre/post triggering of xPC scope.
% tgscopedemo     - Demonstration of xPC TargetScope.
% parsweepdemo    - Demonstrates parameter updates in xPC.
% dataloggingdemo - Demonstrates time- and value-equidistant data logging.
%
% xpcdngdemo      - How to use the demo Dials & Gauges model with xPC Target
% xpcspectrumdemo - How to use the Spectrum Analyzer demo with xPC Target
% xpcbench        - Execute xPC Target benchmarks and show result
%
% xpcasynctrans   - Model demonstrating asynchronous event support and
%                   asynchronous rate transition block
% xpcasyncbuffer  - Model demonstrating asynchronous event support and
%                   asynchronous double buffer blocks
%
% xpccanisa       - Loopback test model for the CAN-AC2 (ISA) board
% xpccanpci       - Loopback test model for the CAN-AC2-PCI (PCI) board
% xpccanpc104     - Loopback test model for the CAN-AC2-104 (PC/104) board
% xpccanintpci    - Loopback test model using asynchronous interrupts
%                   for the CAN-AC2-PCI (PCI) board
% xpccanintpc104  - Loopback test model using asynchronous interrupts
%                   for the CAN-AC2-104 (PC/104) board
% xpccanpcififo1  - Model using FIFO drivers for the CAN-AC2-PCI (PCI) board
% xpccanpcififo2  - Model using FIFO drivers for the CAN-AC2-PCI (PCI) board
% xpccanpcififo3  - Model using FIFO drivers for the CAN-AC2-PCI (PCI) board
% xpccanpcififo4  - Model using FIFO drivers for the CAN-AC2-PCI (PCI) board
% xpccanpcififo5  - Model using FIFO drivers for the CAN-AC2-PCI (PCI) board
% xpccan104fifo1  - Model using FIFO drivers for the CAN-AC2-104 (PC/104) board
% xpccan104fifo2  - Model using FIFO drivers for the CAN-AC2-104 (PC/104) board
% xpccan104fifo3  - Model using FIFO drivers for the CAN-AC2-104 (PC/104) board
% xpccan104fifo4  - Model using FIFO drivers for the CAN-AC2-104 (PC/104) board
% xpccan104fifo5  - Model using FIFO drivers for the CAN-AC2-104 (PC/104) board
%
% xpc8audiochannels - Demonstrates the setup for the AudioPMC with all 8 channels
% xpcFrameloop    - Demonstrates how to use a for loop to iterate through
%                   a frame one sample at a time when the minimum
%                   sampletime is the frame completion time.
%
% xpcUEIFrame     - Demonstrates how to use the frame based acquisition
%                   mode for the UEI MF and MFS boards.
% xpcUEIMasterSlaveFrame - Demonstrates how to use two UEI boards
%                   synchronized as master and slave.
% xpcUEIasync     - Demonstrates the use of an asynchronous interrupt
%                   handler for frame based acquisition at a rate
%                   different than the model sampletime.
% xpcUEIDualasync - Demonstrates the use of more than one asynchronous
%                   interrupt handler with multiple UEI boards doing
%                   frame based acquisition at possibly different rates.


%   Copyright 1996-2002 The MathWorks, Inc.
%   $Revision: 1.5.2.1 $ $Date: 2003/11/20 11:58:54 $
```

DISCLAIMER: These document is just a collection of commands and Help contents of xPC Target version in R14. It could content errors or spelling mistakes, The MathWorks isn't responsible for maluse of commands and results of them which are listed here.
For furthermore support on xPC Target usage, please read the User's Guide or contact the technical Support.