NAME

     mtx-context - ConTeXt Process Management

SYNOPSIS

     mtxrun --script context [*OPTIONS ...* **] [** *FILENAMES* **]**

DESCRIPTION

     ConTeXt Process Management

OPTIONS: BASIC

     --run   process (one or more) files (default action)

     --make create context formats

     --ctx=name

          use ctx file (process management specification)

     --noctx ignore ctx directives and flags

     --interface

          use specified user interface (default: en)

     --autopdf

          close pdf file in viewer and start pdf viewer afterwards

     --purge

          purge files either or not after a run (--pattern=...)

     --purgeall

          purge all files either or not after a run (--pattern=...)

     --usemodule=list

          load the given module or style, normally part of the distribution

     --environment=list

          load the given environment file first (document styles)

     --mode=list

          enable given the modes (conditional processing in styles)

     --path=list

          also consult the given paths when files are looked for

     --arguments=list

          set variables that can be consulted during a run (key/value pairs)

     --randomseed=number

          set the randomseed

     --result=name

          rename the resulting output to the given name

     --trackers=list

          set tracker variables (show list with --showtrackers)

     --directives=list

          set directive variables (show list with --showdirectives)

--silent=list
>
> disable logcategories (show list with --showlogcategories)

--strip   strip Lua code (only meant for production where no errors are expected)

--errors=list
>
> show errors at the end of a run, quit when in list (also when --silent)

--htmlerrorpage
>
> generate html error page instead (optional: =scite)

--noconsole
>
> disable logging to the console (logfile only)

--nodummy
>
> don't create a dummy file (can confuse pdf viewers that keep file in view)

--purgeresult
>
> purge result file before run

--find=string
>
> search for the given pattern in the documentation sources

--forcexml
>
> force xml stub

--forcecld
>
> force cld (context lua document) stub

--forcelua
>
> force lua stub (like texlua)

--forcemp
>
> force mp stub

--arrange
>
> run extra imposition pass, given that the style sets up imposition

--noarrange
>
> ignore imposition specifications in the style

--pages   report pagenumbers of names pages and list references (--page=... or --page=...)

--jit   use luajittex with jit turned off (only use the faster virtual machine)

--jiton   use luajittex with jit turned on (in most cases not faster, even slower)

--once   only run once (no multipass data file is produced)

--runs   process at most this many times

--forcedruns
>
> process this many times (permits for optimization trial runs)

--batchmode
>
> run without stopping and do not show messages on the console

--nonstopmode
>
> run without stopping

--nosynctex
> never initializes synctex (for production runs)

--synctex
> run with synctex enabled (better use upsynctex[state=start])

--nodates
> omit runtime dates in pdf file (optional value: a number (this 1970 offset time) or string "YYYY-MM-DD HH:MM")

--nocompression
> forcefully turns off compression in the backend

--trailerid
> alternative trailer id (or constant one)

--generate
> generate file database etc. (as luatools does)

--paranoid
> do not descend to .. and ../..

--version
> report installed context version

--global
> assume given file present elsewhere

--nofile use dummy file as jobname

--forceinput
> use this file as input file instead of the given one

--expert
> show some more options

OPTIONS: EXPERT

--touch update context version number (also provide --expert, optionally provide --basepath)

--nostatistics
> omit runtime statistics at the end of the run

--profile
> profile job (use: mtxrun --script profile --analyze)

--timing
> generate timing and statistics overview

--keeptuc
> keep previous tuc files (jobname-tuc-[run].tmp)

--keeppdf
> keep previous pdf files (jobname-pdf-keep.tmp)

--keeplog
> keep previous log files (jobname-log-[run].tmp)

--notuc don't write a tuc file

--lmtx   force lmtx mode (when available)

--overloadmode=error|warning|0--6|255
        enable csname overload checking

--extra=name
        process extra (mtx-context-... in distribution)

--extras
        show extras

--ownerpassword
        encrypt the (pdf) file using this master password

--userpassword
        use an additional password for opening the document

--permissions
        list of: print, modify, extract, add, fillin, assemble, quality

OPTIONS: SPECIAL
--pdftex
        process file with texexec using pdftex

--xetex   process file with texexec using xetex

--mkii   process file with texexec

--pipe   do not check for file and enter scroll mode (--dummyfile=whatever.tmp)

--parallel
        process multiple files at the same time (you can specify a number of processes, default:
        8)

--parallellist
        process commands in this file (lines that start with 'context')

--sandbox
        process file in a limited environment

--addbinarypath
        prepend the (found) binarypath to runners

AUTHOR
        More information about ConTeXt and the tools that come with it can be found at:

maillist: ntg-context@ntg.nl / http://www.ntg.nl/mailman/listinfo/ntg-context

webpage: http://www.pragma-ade.nl / http://tex.aanhet.net

wiki: http://contextgarden.net