Contents

Chapter 1

Appendix A: Install and Build ROOT

1.1 License

ROOT is made available under the LGPL v2.1 license. For full details see the file LICENSE in the ROOT distribution.

1.2 Installing ROOT

To install ROOT you will need to go to the ROOT website at: http://root.cern.ch/drupal/content/downloading-root

You have a choice to download the binaries or the source. The source is quicker to transfer since it is only 31 MB, but you will need to compile and link it. The binaries range from 50 MB to 100 MB depending on the target platform.

1.3 Choosing a Version

The ROOT developers follow the principle of "release early and release often", however a very large portion of a user base requires a stable product therefore generally three versions of the system is available for download - new, old and pro:

- The *new* version evolves quickly, with weekly or bi-weekly releases. Use this to get access to the latest and greatest, but it may not be stable. By trying out the new version you can help us converge quickly to a stable version that can then become the new pro version. If you are a new user we would advice you to try the new version.
- The *pro* (production) version is a version we feel comfortable with to exposing to a large audience for serious work. The change rate of this version is much lower than for the new version, it is about 3 to 6 months.
- The *old* version is the previous pro version that people might need for some time before switching the new pro version. The old change rate is the same as for pro.

1.4 Installing Precompiled Binaries

The binaries are available for downloading from http://root.cern.ch/drupal/content/downloading-root. Once downloaded you need to unzip and de-tar the file. For example, if you have downloaded ROOT v5.30 for Linux-SLC5:

```
% gunzip root_v5.30.00.Linux-slc5-gcc4.3.tar.gz
% tar xvf root_v5.30.00.Linux-slc5-gcc4.3.tar
```

This will create the directory root. Before getting started read the file README/README. Also, read the Introduction chapter for an explanation of the directory structure.

1.5 Installing from the Source

The ROOT's source code sits in the GitHub repository https://github.com/root-project/root all the details and options to buld ROOT from sources are given in https://root.cern/building-root

In short, the steps to build ROOT from the sources are the following:

1. Clone the repository:

   ```
   $ git clone https://github.com/root-project/root.git
   ```

2. Make a directory for building

   ```
   $ mkdir build
   $ cd build
   ```

3. Setup and run ROOT

   ```
   $ source bin/thisroot.sh
   $ root
   ```

1.5.0.1 Staying up-to-date

To keep your local ROOT source up-to-date with the GitHub repository you should regularly run the command:

```
% git pull
```

1.6 File system.rootrc

ROOT Environment settings are handled via the class `TEnv`. `gEnv->Print()`shows which values are active. Any settings can be obtained by `TEnv::GetValue` methods that return an integer, double or character value for a named resource. If the resource is not found, the default value (given as the second parameter) is returned.

```
fShowEventStatus = gEnv->GetValue("Canvas.ShowEventStatus",kFALSE);
```

Via the method `TEnv::SetValue` allows you can set the value of a resource or create a new resource:

```
gEnv->SetValue("Root.ObjectStat",1);
```

Path used by dynamic loader to find shared libraries and macros. Paths are different for Unix and Windows. The example shows the defaults for all ROOT applications for either Unix or Windows:

```
Unix.*.Root.DynamicPath:    .:$(ROOTSYS)/lib
Unix.*.Root.MacroPath:      .:$(ROOTSYS)/macros
WinNT.*.Root.DynamicPath:   .;$(ROOTSYS)/bin;$(PATH)
WinNT.*.Root.MacroPath:     .;$(ROOTSYS)/macros
```

Path where to look for `TrueType` fonts:

```
Unix.*.Root.UseTTFonts:     true
*.*.Root.TTFontPath:        $(ROOTSYS)/fonts
```

Use `Net* API` functions:

```
WinNT.UseNetAPI:            true
```

Use thread library (if exists).

```
Unix.*.Root.UseThreads:     false
```

Select the compression algorithm (0=old zlib, 1=new zlib). Setting this to '0' may be a security vulnerability.

```
Root.ZipMode:               1
```

Show where item is found in the specified path:

```
Root.ShowPath:              false
```

Activate memory statistics (`size` and `cnt` are used to trap allocation of blocks of a certain `size` after `cnt` attempts).

```
Root.MemStat:               0
Root.MemStat.size:         -1
Root.MemStat.cnt:          -1
Root.ObjectStat:            0
```

Activate memory leak checker (use in conjunction with `$ROOTSYS/bin/memprobe`). Currently only works on Linux with gcc.

```
Root.MemCheck:               0
Root.MemCheckFile:           memcheck.out
```

Global debug mode. When `>0` turns on progressively more details debugging.

```
Root.Debug:                  0
Root.Stacktrace:             yes
```

Settings for X11 behaviour.

```
X11.Sync:                    no
X11.FindBestVisual:          yes
```

Default editor in use.

```
Unix.*.Editor:               vi
WinNT.*.Editor:              notepad
```

Default 3d Viewer. By default 3-D views are shown in the pad, if the next line is activated, the default viewer will be OpenGL.

```
Viewer3D.DefaultDrawOption: ogl
```

Default Fitter (current choices are `Minuit`, `Minuit2`, `Fumili` and `Fumili2`).

```
Root.Fitter:                 Minuit
```

Specify list of file endings which **`TTabCom`** (TAB completion) should ignore.

```
TabCom.FileIgnore:           .cpp:.h:.cmz
```

1.6.1 TCanvas Specific Settings

Opaque move and resize show full pad during the operation instead of only the outline. Especially for resize you will need serious CPU power. `UseScreenFactor=true` means to size canvas according to size of screen, so a canvas still looks good on a low resolution laptop screen without having to change canvas size in macros.

```
Canvas.MoveOpaque:           false
Canvas.ResizeOpaque:         false
Canvas.UseScreenFactor:      true
```

Hight color 2 is the red one.

```
Canvas.HighLightColor:       2
```

Next three settings are related to different user interface parts of canvas window. If they are set to true, the corresponding event status bar, tool bar, graphics editor to beactivated by default.

```
Canvas.ShowEventStatus:      false
Canvas.ShowToolBar:          false
Canvas.ShowEditor:           false
```

AutoExec allows **`TExec`** objects to be executed on mouse and key events.

```
Canvas.AutoExec:             true
```

Canvas print directory is set to the current one by default:

```
Canvas.PrintDirectory      .
```

Printer settings:

```
WinNT.*.Print.Command:     AcroRd32.exe
#Unix.*.Print.Command:     a2ps -P%p --landscape --columns=2 --margin=30 -rf8.0 %f
Print.Printer:             32-rb20-hp
Print.Directory:           .
Print.FileType:            pdf
```

Default histogram binnings used by `TTree::Draw()` method.

```
Hist.Binning.1D.x:           100
Hist.Binning.2D.x:            40
Hist.Binning.2D.y:            40
Hist.Binning.2D.Prof:        100
Hist.Binning.3D.x:            20
Hist.Binning.3D.y:           20
Hist.Binning.3D.z:           20
Hist.Binning.3D.Profx:      100
Hist.Binning.3D.Profy:      100
```

Default statistics names used for parameters in **`TPaveStats`**:

```
Hist.Stats.Entries          Entries
Hist.Stats.Mean             Mean
Hist.Stats.MeanX            Mean x
Hist.Stats.MeanY            Mean y
Hist.Stats.RMS              RMS
Hist.Stats.RMSX             RMS x
Hist.Stats.RMSY             RMS y
Hist.Stats.Underflow        Underflow
Hist.Stats.Overflow         Overflow
Hist.Stats.Integral         Integral
Hist.Stats.Skewness         Skewness
Hist.Stats.SkewnessX        Skewness x
Hist.Stats.SkewnessY        Skewness y
Hist.Stats.Kurtosis         Kurtosis
Hist.Stats.KurtosisX        Kurtosis x
Hist.Stats.KurtosisY        Kurtosis y
```

1.6.2 THtml Specific Settings

See the reference guide documentation of **`THtml`** class at http://root.cern.ch/root/htmldoc/THtml.html for more details.

XHTML content charset (see http://www.w3.org/TR/2002/REC-xhtml1-20020801, default: ISO-8859-1) is set by:

```
Root.Html.Charset:
```

Stem of a search engine for the documentation, where `%s` is replaced by the term entered in the search text box (example: `http://www.google.com/search?q=%s+site%3Aroot.cern.ch%2Froot%2Fhtml`, default is `""`)

```
Root.Html.Search:
```

Link to the site's search engine (default: `""`, example: `http://root.cern.ch/root/Search.phtml`)

```
Root.Html.SearchEngine:
```

String to prepend to `TClass::GetImplFileName()` names containing directories when looking for source files (default: `""`, example: `../root`)

```
Root.Html.SourcePrefix:
```

Link stem to `ViewCVS` entry for classes, where a class name is assumed to match a file name (default: `""`, example: `http://root.cern.ch/viewcvs`).

```
Root.Html.ViewCVS:
```

Stem of the CERN XWho system (default: `http://consult.cern.ch/xwho/people?`)

```
Root.Html.XWho:
```

If set to Doc++, allow method documentation in front of method even for methods in the source file (default: `""`)

```
Root.Html.DescriptionStyle:
```

Search path for the source and header files with their default settings:

```
Unix.*.Root.Html.SourceDir:  .:src:include
WinNT.*.Root.Html.SourceDir: .;src;include
```

URL stem for ROOT documentation pages (default is `""`).

```
Root.Html.Root:               http://root.cern.ch/root/html
```

Filesystem output directory for generated web pages (default: `htmldoc)`.

```
Root.Html.OutputDir:         htmldoc/
```

Address of the package's home page (default: http://root.cern.ch):

```
Root.Html.HomePage:
```

Location of user defined header and footer files, see http://root.cern.ch/root/html/THtml#conf:header (defaults are `""`, example: `../header.txt`, `../footer.txt`):

```
Root.Html.Header:
Root.Html.Footer:
```

Tag for detecting class description comments (default value is set below).

```
Root.Html.Description:      //____________________
```

Tag for detecting "Author" comment (default value is set below).

```
Root.Html.Author:           // Author:
```

Tag for detecting "last updated" comment. `THtml` uses the current date if this tag is not found in a class source file (default value is set below).

```
Root.Html.LastUpdate:       // @(#)
```

Tag for detecting "Copyright" comment (default value is set below).

```
Root.Html.Copyright:        * Copyright
```

1.6.3 GUI Specific Settings

Set the "`native`" ROOT GUI interface to be used in a ROOT session. If set to "`qt`", the "`native`" GUI interface is replaced with one based on Qt by the regular ROOT plug-in mechanism.

```
Gui.Backend:                native
Gui.Factory:                native
```

GUI default fonts in use:

```
Gui.DefaultFont:            -adobe-helvetica-medium-r-*-*-12-*-*-*-*-*-iso8859-1
Gui.MenuFont:               -adobe-helvetica-medium-r-*-*-12-*-*-*-*-*-iso8859-1
Gui.MenuHiFont:             -adobe-helvetica-bold-r-*-*-12-*-*-*-*-*-iso8859-1
Gui.DocFixedFont:           -adobe-courier-medium-r-*-*-12-*-*-*-*-*-iso8859-1
Gui.DocPropFont:            -adobe-helvetica-medium-r-*-*-12-*-*-*-*-*-iso8859-1
Gui.IconFont:               -adobe-helvetica-medium-r-*-*-10-*-*-*-*-*-iso8859-1
Gui.StatusFont:             -adobe-helvetica-medium-r-*-*-10-*-*-*-*-*-iso8859-1
```

Regular background and foreground colors in use:

```
Gui.BackgroundColor:         #c0c0c0
Gui.ForegroundColor:         black
```

Selection background and foreground colors in use:

```
Gui.SelectBackgroundColor:   #000080
Gui.SelectForegroundColor:   white
```

Document background and foreground colors in use:

```
Gui.DocumentBackgroundColor: white
Gui.DocumentForegroundColor: black
```

Tooltip background and foreground colors in use:

```
Gui.TooltipBackgroundColor:  LightYellow
Gui.TooltipForegroundColor:  black
```

Path where all GUI icons in use can be found:

```
Gui.IconPath:                $(HOME)/icons:$(ROOTSYS)/icons:.
```

Mime type file setting:

```
Gui.MimeTypeFile:            $(HOME)/.root.mimes
```

If `$(HOME)/.root.mimes` does not exists, defaults to this:

```
#Gui.MimeTypeFile:            $(ROOTSYS)/etc/root.mimes
```

1.6.4 TBrowser Settings

Current icon style selection - can be either `small`, `big`, `list`, `details`:

```
Browser.IconStyle:           small
```

Current sorting rule applied on the browser objects - can be `name`, `type`, `size`, `date`:

```
Browser.SortBy:              name
```

Number of items in a group view:

```
Browser.GroupView:           10000
```

Show or not hidden items:

```
Browser.ShowHidden:          no
```

Create a thumbnail view after executing the macro (default is `yes`).

```
Browser.AutoThumbnail:       yes
```

1.6.5 TRint Specific Settings

Rint (interactive ROOT executable) specific alias, logon and logoff macros.

```
Rint.Load:               rootalias.C
Rint.Logon:              rootlogon.C
Rint.Logoff:             rootlogoff.C
```

Record ROOT session commands in a given history file (default is `$(HOME)/.root_hist`). If set to "`-`", it turn off the command recording.

```
Rint.History:            $(HOME)/.root_hist
```

Next two lines set the history file size handling. Once `HistSize` is reached, the last `HistSave` entries are removed. If `HistSize` is set to 0, it turns off command recording. Both values can be overridden by environment variable `ROOT_HIST=size[:save]`, where the "`:save`" part is optional.

```
Rint.HistSize:         500
Rint.HistSave:         400
```

1.6.6 ACLiC Specific Settings

`ACLiC.Linkdef` specifies the suffix that will be added to the script name to try to locate a custom linkdef file when generating the dictionary.

```
ACLiC.Linkdef:          _linkdef
```

The top directory for storing the libraries produced by ACLiC is set by:

```
ACLiC.BuildDir:        /where/I/would/like/my/compiled/scripts
```

The additional include directives for ACLiC compilations are set by:

```
ACLiC.IncludePaths:     -I/where/the/includes/are
```

1.6.7 PROOF Related Variables

PROOF debug options.

```
Proof.DebugLevel: 0
Proof.DebugMask:-1
```

PROOF GDB hooks allows a debugger to be attached early in the startup phase of `proofserv`:0 - don't wait; 1 - master proofserv enters wait loop; 2 - slave proofserv enters wait loop; 3 - any proofserv enters wait loop

```
Proof.GdbHook:     0
```

On the master to enable the parallel startup of workers using threads set next to "`yes`" (default is "`no`"):

```
Proof.ParallelStartup: no
Proof.StatsHist:       no
Proof.StatsTrace:      no
Proof.SlaveStatsTrace: no
```

```
Proof.CondorHome:     /opt/condor
Proof.CondorConfig:   /opt/condor/etc/condor_config
```

```
PEAC.GmUrl:           http://somewhere:8080/clarens/
PEAC.LmUrl:           http://elsewhere:8080/clarens/
```

Certificate and key

```
Clarens.CertFile:     $(HOME)/.globus/usercert.pem
Clarens.KeyFile:      $(HOME)/.globus/userkey.pem
```

1.6.7.1 Settings Related to Authentication for rootd and proofd

Default authentication method for `rootd` and `proofd`. These are supported for backward compatibility but have a very low priority. System defaults are generated by configure as a list in `system.rootauthrc` in `$ROOTSYS/etc/` or `/etc/root`; the file `$HOME/.rootauthrc` can be used to override the system defaults.

The value meaning: `0=UsrPwd`, `1=SRP`, `2=Krb5`, `3=Globus`,`4=SSH`, `5=UidGid`.

```
Rootd.Authentication:    0
Proofd.Authentication:   0
```

Connection is shutdown at timeout expiration. Timeout is in seconds. Negotiation cannot be attempted at low level (i.e. inside TAuthenticate::Authenticate()) because of synchronization problems with the server. At higher level, TAuthenticate::HasTimedOut() gives information about timeout: 0 = no timeout; 1 = timeout, no methods left; 2 = timeout, still methods to be tried. Caller should decide about an additional attempt. Timeout is disabled by default (< 0). It can be changed on-the-fly with the method `TAuthenticate::SetTimeOut(to_value)`.

```
Auth.Timeout:            -1
```

Password dialog box is set to 0 if you do not want a dialog box to be popped-up when a password is requested. Default setting is 1.

```
Auth.UsePasswdDialogBox: 0
```

Set the following to 1 if you want full SRP authentication in PROOF (Client-to-Master and Master-to-Slave).

```
Proofd.SendSRPPwd:         0
```

Set next to 1 to use SSH authentication in PROOF servers (Master-to-Slave or Slaves-to-DataServers). This is switched off by default because credentials forwarding for SSH is not controlled by the system; however the user may have other ways to guarantee it, so it may want to switch it on.

```
ProofServ.UseSSH:        0
```

Default login name (if not defined it is taken from `$(HOME)`).

```
UsrPwd.Login:            qwerty
SRP.Login:               qwerty
Krb5.Login:              qwerty@LOCAL.DOM.AIN
Globus.Login:            cd:~/.globus cf:usercert.pem  kf:userkey.pem
ad:/etc/grid-security/certificates
SSH.Login:               qwerty
UidGid.Login:            qwerty
```

To be prompted for login information.

```
UsrPwd.LoginPrompt:      yes
SRP.LoginPrompt:         yes
Krb5.LoginPrompt:        yes
Globus.LoginPrompt:      yes
SSH.LoginPrompt:         yes
UidGid.LoginPrompt:      yes
```

To reuse established security context.

```
UsrPwd.ReUse:            yes
SRP.ReUse:               no
```

```
Krb5.ReUse:             no
Globus.ReUse:           yes
SSH.ReUse:              yes
```

Duration validity of the sec context for UsrPwd, SRP and SSH. Format: <hours>:<minutes> (default 24:00)

```
#UsrPwd.Valid:          24:00
#SRP.Valid:             24:00
#SSH.Valid:             24:00
```

To control password encryption for UsrPwd authentication.

```
UsrPwd.Crypt:             yes
```

Globus Miscellaneous - Globus Proxy duration: `HH:MM` (ex `12:15` for 12 hours and 15 min); '`default`' for system default.

```
Globus.ProxyDuration:     default
#Globus.ProxyDuration:    12:15
```

Number of bits for the initial key.

```
Globus.ProxyKeyBits:      1024
```

Path to alternative '`ssh`' (to override `$PATH` if ever needed).

```
SSH.ExecDir:              /usr/bin
```

In case of error, `SSH` returns 1 (or `256=0x100`). To trap those errors for which one should retry, error printouts must be parsed; any substring found under the **`TEnv`** `SSH.ErrorRetry` triggers a retry condition; strings can be added here in the form (including double quotes):

```
+SSH.ErrorRetry:         "<error_string>"
```

This is what one usually gets if the server has reached the maximum number of sshd daemons (defined by `MaxStartups` in `sshd_config`); this is a typical case in which one should retry.

```
SSH.ErrorRetry:          "Connection closed by remote host"
```

Max number of retries for SSH in case of retry error (see above).

```
SSH.MaxRetry:            100
```

Type of key to be used for RSA encryption: `0=local`; `1=SSL` (`default` if `openssl` available).

```
RSA.KeyType:             1
```

In case of '`RSA.KeyType: 1`' this specifies the number of bits to be used for the Blowfish key used to encrypt the exchanged information: default - 256, minimum - 128, maximum - 15912.

```
SSL.BFBits:              256
```

1.6.7.2 Server Authentication in TServerSocket

General setting: file with server access rules

```
SrvAuth.DaemonRc:        /etc/root/system.daemonrc
```

Check of host equivalence via `/etc/hosts.equiv` or `$HOME/.rhosts`.

```
SrvAuth.CheckHostsEquivalence: 1
```

SRP: pass file (default `$HOME/.srootdpass`).

```
SrvAuth.SRPpassfile:     $HOME/.srootdpass
```

Globus/GSI: `hostcert` configuration file.

```
SrvAuth.HostCert:         /etc/root/hostcert.conf
```

Globus/GSI: `gridmap` file.

```
SrvAuth.GridMap:          /etc/grid-security/grid-mapfile
```

SSH: port for the `sshd` daemon.

```
SrvAuth.SshdPort:         22
```

Force file opening via **`TNetFile`** (**`TXNetFile`**) if a hostname is specified in the Url. By default, for local files `TFile::Open()` invokes directly **`TFile`.**

```
TFile.ForceRemote:        yes
```

Special cases for the **`TUrl`** parser, where the special cases are parsed in a protocol + file part, like rfio:host:/path/file.root, castor:/path/file.root or /alien/path/file.root. In case the file namespace descriptor ends with - the namespace is not a part of the filename. Extend in private .rootrc with a +Url.Special line.

```
Url.Special:              file: rfio: hpss: castor: gfal: dcache:
+Url.Special:             /alien/- /castor/
```

1.6.7.3 PROOF XRD Client Variables

Debug level (if <=0 : none, 1 : low, 2 : medium, 3 : high)

```
XProof.Debug:       0
```

Socket read timeout [in secs: default 10 secs]

```
XProof.ReadTimeout: 10
```

The following env vars are handled by **`TXNetFile`** and related classes (module `netx`, `libNetx.so`).

`XNet.ConnectTimeout` - maximum time to wait before server's response on a connect [10 s]

`XNet.RequestTimeout` - maximum time to wait before considering a read/write failure [60 s]

`XNet.ConnectDomainAllowRE` - sequence of **`TRegexp`** regular expressions separated by a `|`. A domain is granted access to for the first connection if it matches one of these regexps. Example:

```
slac.stanford.edu|pd.infn.it|fe.infn.it
```

`XNet.ConnectDomainDenyRE` - sequence of TRegexp regular expressions separated by a `|`. A domain is denied access to for the first connection if it matches one of these regexps.

`XNet.RedirDomainAllowRE` - sequence of TRegexp regular expressions separated by a `|`. A domain is granted access to for a redirection if it matches one of these regexps. Example:

`XNet.RedirDomainDenyRE` - sequence of TRegexp regular expressions separated by a `|`. A domain is granted access to for a redirection if it matches one of these regexps.

`XNet.MaxRedirectCount` - maximum number of redirections from server [default - 255]

`XNet.Debug` - log verbosity level (0=nothing,1=messages of interest to the user, 2=messages of interest to the developers (includes also user messages), 3=dump of all sent/received data buffers (includes also user and developers messages). [default - 0]

`XNet.ReconnectTimeout` - sleep-time before going back to the load balancer (or rebouncing to the same failing host) after a read/write error [default - 10s]

`XNet.StartGarbageCollectorThread` - for test/development purposes. Normally nonzero (true), but as workaround for external causes someone could be interested in not having the garbage collector thread around. [experimental!]

`XNet.GoAsynchronous` - default is 0. When activated, **`XTNetFile`** works in async mode, allowing input buffering and unsolicited responses [experimental!]

`XNet.TryConnect` - Number of tries connect to a single server before giving up.

`XNet.TryConnectServersList` - number of connect retries to the whole server list given [default - 240]

`XNet.PrintTAG` - Print a particular string the developers can choose to quickly recognize the version at run time [default - 0]

Example of custom setting for the Rint application (root.exe). This overrides the default specified above for a generic application. Color 5 is yellow.

```
Rint.Canvas.HighLightColor:      5
```

1.7 Documentation to Download

- The latest ROOT Users Guide

- https://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuide.html
- ROOT Reference Guide
- http://root.cern.ch/root/Reference.html