13 DATE installation guide

This chapter describes how to retrieve the DATE software from the reachieve and how to install DATE on the target systems.

13.1 Hardware and software platforms

The system supports the platforms listed in table 13.1.

Table 13.1 Hardware and software platforms

Hardware	Operating system	LDC	GDC	Monitoring	Run control
Motorola MVME 2600	AOS	✓	✓	✓	✓
IBM 43P	AIX	✓	✓	✓	✓
SUN SPARC	Solaris	✓	✓	✓	✓
HP	HP-Unix			✓	
PC	Windows 95				✓
PC	Linux	✓	✓	✓	✓
Compaq Alpha	OSF-1		✓	✓	

13.2 Getting the software

The distribution kits of DATE are accessible from the web. An authorization is required.

The kit repository is located at the following URL:

http://aldwww.cern.ch/Collaborators/DATE/Releases

There are two types of kits, called respectively ***core*** and ***full*** (this is indicated in the file name).

1. The ***full*** kit contains the DATE distribution and the distribution of Java. Its size is around 80 MB. It should be used for the first installation and, in the successive installations, when the Java version has changed. Since several DATE packages are written in Java, it is important to make sure that the Java release on the machine is in step with the one used by DATE
2. The ***core*** kit contains exclusively the DATE distribution. Its size is around 15 MB, therefore it is faster to retrieve. It may be used for upgrading the DATE versions within a major release.

The name of the kit contains the date of generation and is suffixed with either ***.tar*** or ***.tar.gz***

Before retrieving the distribution kit on your file system (e.g. on ***/tmp***) make sure that the ***/tmp*** directory has the read / write / execute protections for everybody, otherwise set the privileges in the following way:

```
> chmod a+rwx /tmp
```

13.3 First time installation

13.3.1 Setting up the file base

1. Login as ***root***.
2. Authorize ***root*** to open a remote shell on all the machines that will be running DATE. This is done by editing (or creating) the ***/.rhosts*** file to contain a line with the name of the machine where you ar logged in.
3. Create the directory where the DATE software should be installed.
   You should call it ***/date***.
   If you choose another name (e.g. */date_root*), you must set up a symbolic link in the following way:

   ```
   > ln -s /date_root /date
   ```

   Such a link must be declared in all the machines in the system.

4. Create a directory where your experiment specific files will reside.
5. Define the symbol ***DATE_SITE*** to point to the new directory above.

   In ***Bourne shell*** do the following:

   ```
   > export DATE_SITE; DATE_SITE=/directory
   ```

   In ***C-shell*** do the following:

   ```
   > setenv DATE_SITE /directory
   ```

   In ***BASH*** do the following:

   ```
   > declare -x DATE_SITE=/directory
   ```

   where *directory* is the name of the new directory.

6. Login with a user account.
7. Expand the distribution kit, previously retrieved in ***/tmp***. All the files will be automatically placed in the right location.

   ```
   > cd /date
   > rm -rf *
   > tar xf /tmp/distribution_kit.tar
   ```

   Removing all the files before invoking ***tar*** is useful to make sure that files no longer necessary are deleted. Be careful not to delete either the directory ***.commonScripts*** or the files in it.
   If you get the compressed version of the kit, suffixed with ***.tar.gz***, you should first de-compress it:

   ```
   > gunzip distribution_kit.tar.gz
   ```

If disk space is a problem, decompression and unpacking can be done in a single step. With /date as your working directory, do:

```
> gunzip -c distribution_kit.tar.gz | tar xf -
```

8. Setup all the DATE symbols:

   In ***Bourne shell*** and in ***BASH*** do the following:

   ```
   > . [ -r /date/setup.sh ] && . /date/setup.sh
   ```

   In ***C-shell*** do the following:

   ```
   > source /date/setup.csh
   ```

9. The operations described in the points 5 and 8 above are necessary each time you want to use DATE. Therefore, you may want to add them to your login file (e.g. in ***.zshenv***, ***.profile*** or ***.bashrc***).

10. Create under ***${DATE_SITE}*** the following sub-directories:

    logFiles
    configurationFiles
    runControl
    eventBuilder/*eb_host*

    where *eb_host* is the name of the machine hosting the event builder.

11. Create under ***${DATE_SITE}*** one directory for each CPU in your system. The name of each directory must be the name of the corresponding machine (e.g. ***mppcal01***)

12. Give write access for the group and for others to all the directories previously created:

    ```
    > chmod -R a+w ${DATE_SITE}
    ```

13. Create the file ***rcShmKey*** onto the site-dependent area:

    ```
    > touch ${DATE_SITE}/runControl/rcShmKey
    ```

    Make sure that ***rcShmKey*** has read access for all. If not, change the access rights:

    ```
    > chmod a+r ${DATE_SITE}/runControl/rcShmKey
    ```

13.3.2 Internet services

Some INETD services are needed by DATE:

- ***date_rcs***
  Run control service, used to start the ***rcServer***. It should be added to all the

machines declared as either LDC or GDC.

- ***date_evb***
  Event builder service, used to start the ***gdcServer***. It should be added to the machines declared as GDC.
- ***date_info***
  Info logger dæmon, used to start the ***infoDaemon***. It should be added to the machine declared as ***DATE_INFOLOGGER_LOGHOST*** in ***${DATE_SITE_CONFIG}/infoLogger.config***.
- ***date_evbd***
  Event builder control service, used to start the ***ebDaemon***. It should be added to the machine declared as GDC.
- ***date_mon***
  Monitoring service, used for hosts who wish to offer monitoring stream via network. It should be added to any machine (online or offline) who wish to offer monitoring streams to the outside world.
- ***date_watchmon***
  Monitoring service, used to monitor the behaviour of the monitoring scheme.

There is a procedure that implements all these services. It must be invoked from ***root***. Make sure that the DATE symbols are still defined, otherwise execute again the procedure described in 13.3.1 point 8. The command to invoke the procedure is the following:

```
> dateNetwork (-c|-i) [-g"GDC1 GDC2 ..."][-l"LDC1 LDC2
..."][-G"InfoLogger1 InfoLogger2 ..."][-m"MON1 MON2 ..."]
```

where:
-c performs the check that the network is properly setup to run DATE (no modifications performed)
-g indicates the list of GDCs
-G indicates the list of machines where the infoLogger may run
-i installs the setup needed by DATE
-l indicates the list of LDCs
-m indicates the list of machines from which event monitoring is allowed

The list of machines must be a blank-separated sequence of hostnames, e.g.:

```
> dateNetwork -c -g"host1 host2 host3" -l"host4 host5"
-Ghost6 -m"host7 host8 host9"
```

All hosts shall be reachable directly by "rsh" and the caller must be authorized to remote shell as user "root" on the remote machines. This is normally the case, if you have followed the instructions in 13.3.1.

The command ***dateNetwork*** performs the following steps.

1. The file ***/etc/inetd.conf*** of each machine where the INETD services are required will contain the code fragment in listing 13.1.

Listing 13.1 INETD service configuration

```
1: #
2: #  DATE service handlers
3: #
4: date_rcs stream tcp nowait nobody /date/runControl/OS/rcServer.sh
   rcServer /date_site
5: date_info stream tcp nowait nobody
   /date/infoLogger/OS/loggerDaemon loggerDaemon
6: date_evb  stream tcp nowait nobody
   /date/eventBuilder/OS/GDCserver.sh GDCserver /date_site
7: date_evbd stream tcp nowait nobody
   /date/eventBuilder/OS/eventBuilderDaemon.sh eventBuilderDaemon
   /date_site
8: date_mon stream tcp nowait nobody /date/monitoring/OS/mpDaemon.sh
   mpDaemon /date_site
9: date_watchmon stream tcp nowait nobody
   /date/watchMon/OS/watchMonServer.sh watchMonServer /date_site
```

Note that the symbol */date_site* represents the real name of the directory indicated by the symbol ***DATE_SITE*** (the symbol ***DATE_SITE*** cannot be used in this context). Conversely, the symbol *OS* points to the machine's operating system, available within the symbol ***DATE_SYS***.

The service defined in line 4 is needed on GDCs and LDCs. The service defined in line 5 is required by the infoLogger host. The services defined in line 6 and 7 are needed on the GDCs. The service defined in line 8 is needed by all machines who wish to offer monitoring streams. The service defined in line 9 is needed by all machines offering monitoring streams whose behaviour has to be monitored via the watchMon utility.

2. The file ***/etc/services*** of each machine - where the INETD services are defined - will contain the code fragment in listing 13.2.

Listing 13.2 INETD services

```
1: #
2: #   DATE servers
3: #
4: date_rcs       6101/tcp  # DATE Run Control Server
5: date_evb       6201/tcp  # DATE Event Builder: GDC server
6: date_info      6301/tcp  # DATE info logger daemon
7: date_evbd      6401/tcp  # DATE Event Builder Daemon
8: date_mon       6601/tcp  # DATE Monitoring server
9: date_watchmon  6611/tcp  # DATE Monitoring watch
```

3. To make the modifications effective, the following command will be invoked:

   On AIX:
   > refresh -s inetd

   On Solaris:
   > kill -HUP inetdPID
   where inetdPID is the process ID of the inetd process (it can be determined via the command:
   > ps -e|grep inetd

   On Linux:
   > kill -HUP 'cat /var/run/inetd.pid'

13.3.3 Run-time libraries

The run control program needs the X11, TCL and TK shearable libraries installed on the host where ***runControl*** is active. The run control agent (***rcSever***) needs the TCL shearable library installed on all the hosts directly involved in the DAQ (LDCs and GDCs).

These libraries are supposed to be available in the standard place ***/usr/local/lib directory***

Usually, this is the result of the operating system installation procedure. If not, the library installation will have to be performed purposely.

13.4 Installation of a new release

The procedure of installing a new release of DATE consists of replacing all the files in the ***${DATE_ROOT}*** area.

1. Import the distribution kit, as described in 13.2.
2. Expand the distribution kit, as described in 13.3.1 point 6.

All the site-specific files are left untouched by the procedure. It is, though, important to check the release notes to see whether the new packages require modification or additions to the configuration files.

13.5 Run control configuration

13.5.1 Run-control windows configuration

The run control presents to the operator a few windows from which it is possible to set up and modify the hardware configuration (by selecting the machines involved in the run and their relationship) and to establish the parameters that define the run conditions.

The content of the run-control windows is itself parametrized. Items to appear on the display windows must previously be defined in the file ***${DATE_SITE_CONFIG}/runControl.config***. This file must be prepared with an editor. An example is shown in Listing 13.3. Another example exists in the distribution: ***/date/runControl/runControl.config***, which may be copied in the right place and edited there.

The file ***runControl.config*** is structured as a sequence of keywords (prefixed with either > or *) followed by a list of definitions of the items related to the keyword. Each item occupies one line and is made of a sequence of parameters.

The syntax rules are the following:

- String parameters must be enclosed within quotes (“).
- Blank lines are ignored.
- Blank characters are used as parameter separators, otherwise they are ignored.
- Lines starting with # are ignored (comment lines).
- Lines starting with either > or * must specify a valid keyword.
- A keyword followed by another keyword is ignored (even if not valid).
- Keywords may be given in any order.

The file is interpreted in one pass. A syntax error stops the interpretation of the file (remaining lines are ignored).

Listing 13.3 Example of the file ***runControl.config***

```
 1:  # Example of configuration file for the runControl program
 2:  # Compliant with DATE v3.2
 3:
 4:  >WINDOWTITLE
 5:  DATE run control
 6:
 7:  >PARFILE
 8:  runControl.params
 9:
10:  >LDC
11:  rsald01 “Front-end 1” 0
12:  rsald02 “Front-end 2” 1
13:  mppcal03 “Front-end 3” 2
14:  mppcal04 “Front-end 4” 3
15:
16:  >GDC
17:  mppcal05 “Event builder”
18:
19:  >RUNPAR
20:  maxEvents “Max. number of events”
21:
22:  >CNFPAR
23:  maxBytes “Max. kBytes to record”
24:  maxEventSize “Max. event size (bytes)”
25:  maxFileSize “Max. file size (kBytes)”
26:  phaseTimeoutLimit “Max. time for SOR/EOR phases (sec)”
27:
28:  >LDCPAR
29:  recordingDevice “Recording device”
30:  monitorEnableFlag “Monitor enable flag”
31:  randEventMinSize “Randev min size”
32:  randEventMaxSize “Randev max size”
33:  randEventInterval “Randev interval”
34:  logLevel “Logging level”
35:
36:  >GDCPAR
37:  ebRecordingDevice “EVB recording device”
38:  monitorEnableFlag “Monitor enable flag”
39:  logLevel “Logging level”
40:
41:  >LDCSTATUS
42:  eventCount “Number of events”
43:  triggerCount “Number of triggers”
44:  eventsTransferred “Events transferred”
45:  bytesTransferred “kBytes transferred”
46:  readoutFlag “SOR/EOR phase”
47:
48:  >GDCSTATUS
49:  eventCount “Number of events”
50:  eventsRecorded “Events recorded”
51:  bytesRecorded “kBytes recorded”
52:  monitorEnableFlag “Monitor enable flag”
53:  fileCount “File count”
54:  bytesInFile “kBytes in file”
55:
56:  >TCLEVAL
57:  set ldcEventBufferSize 1024000
58:  source /date/runControl/bufferStatus.tcl
```

Valid keywords are the ones shown in table 13.2.

Table 13.2 Keywords of the configuration file

Keyword	Following items
COMMENT	Catch all: all the following lines are ignored, until the next keyword.
WINDOWTITLE	The title of the run-control program main window.
PARFILE	The directory and file name of the run control parameters. If specified, the run control program will try to read the parameters from this file and will propose this file as default location for further parameter saving and restoring.
LDC	The list of the machines where the front-end software runs. Each item is the name of the machine followed by a label which will appear in the run control main window. The last parameter is the detector bit which will be written in the detector mask field of the event header. Valid detector bits are 0 up to 126. This list contains all the machines among which the active machines will be chosen.
GDC	The list of the machines where the event builder and the monitor process should be launched at start of run. Each item is the name of the machine followed by a label which will appear in the run control main window. This list contains all the machines among which the active machines will be chosen.
RUNPAR	The list of control parameters that will appear in the main window, under the heading ***Run parameters***. These parameters will be set at start of run in the shared memory segment of the run control of all the machines (LDCs and GDCs). Each item is the conventional name of the parameter followed by a label which will appear in the main window.
CNFPAR	The list of control parameters that will appear in the window labelled ***Configuration parameters***, under the heading ***Common parameters***. The treatment is the same as for the RUNPAR parameters; the only difference is the location on the screen, which implies less visibility for the operator.
LDCPAR	The list of control parameters which will appear in the LDC part of the window labelled ***Configuration parameters***. These parameters will be set at start of run in the shared memory segment of the run control of all the LDCs. Each item is the conventional name of the parameter followed by a label which will appear in the parameter window.
GDCPAR	The list of control parameters which will appear in the GDC part of the window labelled ***Configuration parameters***. These parameters will be set at start of run in the shared memory segment of the run control of all the GDCs. Each item is the conventional name of the parameter followed by a label which will appear in the parameter window.
LDCSTATUS	The list of control parameters which will be displayed in the LDC part of the window labelled ***Status display***. Each item is the conventional name of the parameter followed by a label which will appear in the display window.

Table 13.2 Keywords of the configuration file

Keyword	Following items
GDCSTATUS	The list of control parameters which will be displayed in the GDC part of the window labelled ***Status display***. Each item is the conventional name of the parameter followed by a label which will appear in the display window.
TCLEVAL	The list of tcl statements to be executed by the run control program. The execution of these statements will happen after the run control has processed the configuration file and is properly configured, and before the optional parameter file is read. The main use is to plug in additional tcl source code.
TCLEARLYEVAL	The list of tcl statements to be executed by the run control program. These statements are immediately passed to the tcl interpreter, line by line. A mistake in the statements will cause an abnormal exit of the run control program, since the proper recovery mechanism is not yet available at this point in time. The main use is to define environment variables necessary to the run control but not available in some environments.

The keywords RUNPAR, CNFPAR, LDCPAR, GDCPAR, LDCSTATUS and GDCSTATUS are followed by the declaration of some control parameters. The list of the parameters is in Table 13.3, with a description of the parameters and the place where they may be declared. The parameter names are conventional and must be used in the declaration.

The complete list of all the parameters used in DATE can be found in the file:

/date/runControl/rcShm.h.

The parameters declared under the keywords CNFPAR, LDCPAR and GDCPAR will appear in the run-control window labeled Configuration parameters (Figure 13.1). These parameters are the ones that are relatively stable and do not need to be changed by the operator in normal production runs.

The parameters declared under the keyword RUNPAR will appear in the main run-control window. These parameters are supposed to be possibly changed from run to run, such as ***maxEvents***.

Table 13.3 Run control parameters

Parameter name	Description	Set by	To appear in window
maxEvents	Maximum number of events to be collected in a run. When a DAQ machine hits the limit, an end of run request is issued. Zero means no limit.	The operator via the run control.	Run control main window - Run parameters (under >RUNPAR in runControl.config).

Table 13.3 Run control parameters

Parameter name	Description	Set by	To appear in window
maxBytes	Maximum number of kBytes to be collected in a run. When a DAQ machine hits the limit, an end of run request is issued. Zero means no limit.	The operator via the run control.	Configuration parameters window - Common parameters (under >CNFPAR in runControl.config).
maxEventSize	Maximum event size in bytes. Used by readout to reserve space in the event buffer. If this number is too large, space is waisted at the end of the buffer. If it is too small, the ***ReadEvent*** routine in ***readList*** may overwrite and corrupt valuable data or code.	The operator via the run control.	Configuration parameters window - Common parameters (under >CNFPAR in runControl.config).
triggerCount	Incremented by readout at each physics event (not for the other types of events) after calling ***ReadEvent***, and then stored in eventHeader.triggerNb.	Readout.	Status display window (under >LDCSTATUS in runControl.config).
eventCount	Incremented by readout for all types of events before calling ReadEvent. It is compared to maxEvents to stop the run. Incremented by the event builder as well.	Readout and event builder.	Status display window (under >LDCSTATUS and >GDCSTATUS in runControl.config).
burstCount	Set by readout at each event, after calling ReadEvent, to the value found in eventHeader.burstNb.	Readout.	Status display window (under >LDCSTATUS in runControl.config).

Table 13.3 Run control parameters

Parameter name	Description	Set by	To appear in window
eventsInBurst-Count	Set by readout at each event, after calling ReadEvent, to the value found in event-Header.nbInBurst.	Readout.	Status display window (under >LDC-STATUS in runControl.config).
burstFlag	Not used by DATE. It may be used to indicate the burst condition, to be passed either from the operator to readout or the other way round.	Either the operator or readout (it depends on the purpose).	It depends on the purpose.
eventsRecorded	Counter of events recorded on disk, either by the LDC (if alone) or by the GDC.	Either readout or event builder.	Status display window (under either >LDCSTATUS or >GDCSTATUS in runControl.config).
bytesRecorded	Counter of kBytes recorded on disk, either by the LDC (if alone) or by the GDC.	Either readout or event builder.	Status display window (under either >LDCSTATUS or >GDCSTATUS in runControl.config).
eventsTransferred	Counter of events transferred from the LDC to the GDC.	Readout.	Status display window (under >LDC-STATUS in runControl.config).
bytesTransferred	Counter of kBytes transferred from the LDC to the GDC.	Readout.	Status display window (under >LDC-STATUS in runControl.config).
readoutFlag	A number indicating the phase of the start of run and end of run procedures. At the beginning of the procedure it is set to 1, then it will run from 1000 to 6000 at start of run, and from 11000 to 17000 at end of run. Zero means completion.	Readout.	Status display window (under >LDC-STATUS in runControl.config).

Table 13.3 Run control parameters

Parameter name	Description	Set by	To appear in window
phaseTimeoutLimit	Maximum duration (in seconds) of any start of run and end of run phase. Suggested value is 30. Used by rcServer to abort the run if readout does not complete the phase in due time.	The operator via the run control.	Configuration parameters window - Common parameters (under >CNFPAR in runControl.config).
randEventMinSize	Optional parameter used exclusively by the example of readout program contained in the DATE distribution kit. Minimum event size for the random generator.	The operator via the run control.	Optional. Configuration parameters window - LDC parameters (under >LDCPAR in runControl.config).
randEventMaxSize	Optional parameter used exclusively by the example of readout program contained in the DATE distribution kit. Maximum event size for the random generator.	The operator via the run control.	Optional. Configuration parameters window - LDC parameters (under >LDCPAR in runControl.config).
randEventInterval	Optional parameter used exclusively by the example of readout program contained in the DATE distribution kit. Time interval between events in microseconds.	The operator via the run control.	Optional. Configuration parameters window - LDC parameters (under >LDCPAR in runControl.config).

Table 13.3 Run control parameters

Parameter name	Description	Set by	To appear in window
maxFileSize	Each run may be recorded on multiple files. This is the maximum size of each file in kBytes. It is used either by the recorder (LDC alone) or by the event builder (GDC).	The operator via the run control.	Configuration parameters window - Common parameters (under >CNFPAR in runControl.config).
bytesInFile	Counter of kBytes recorded in the current recording file. It is set either by the recorder (LDC alone) or by the event builder (GDC).	Either readout or event builder.	Status display window (under either >LDCSTATUS or >GDCSTATUS in runControl.config).
fileCount	Counter of number of files recorded in the current run. It is set either by the recorder (LDC alone) or by the event builder (GDC)	Either readout or event builder.	Status display window (under either >LDCSTATUS or >GDCSTATUS in runControl.config).
recordingDevice	A character string indicating the recording device of an LDC. It may be either a file name or a machine name (the latter followed by :). It may be a list of destinations if there are more than one GDC (see “3.1.1” on page 22.).	The operator via the run control.	Configuration parameters window - LDC parameters (under >LDCPAR in runControl.config).
ebRecordingDevice	A character string indicating the recording device of an GDC. It must be a file name.	The operator via the run control.	Configuration parameters window - GDC parameters (under >GDCPAR in runControl.config).

Table 13.3 Run control parameters

Parameter name	Description	Set by	To appear in window
monitorEnable-Flag	Switch to enable and disable the possibility of monitoring from a given machine. Zero (0) means disabled, one (1) enabled.	The operator via the run control.	Configuration parameters window - LDC and GDC parameters (under both >LDC-PAR and >GDC-PAR in runControl.config).
recorderSleep-Time	The recorder goes to sleep while events are arriving (if enableRecordingInBurst is disabled), to give priority to readout.The time interval (expressed in microsec.) is picked up from this parameter. Default is tuned to 10 millisec.	Optional (the default is well tuned). The operator via the run control.	Optional. Configuration parameters window - LDC parameters (under >LDCPAR in runControl.config).
enableRecording-InBurst	Switch to enable and disable the recorder during a burst of events. Zero (0) means disabled, one (1) enabled. Default is disabled, which is adapted to fixed target experiments (with bursts). Collider experiments (continuous beam) should enable it.	Optional. The default is adapted to fixed target experiments (with bursts). The operator via the run control.	Optional. Configuration parameters window - LDC parameters (under >LDCPAR in runControl.config).
logLevel	It controls the generation of messages by all the date processes running on a DAQ machine.	The operator via the run control.	Configuration parameters window - LDC and GDC parameters (under both >LDC-PAR and >GDC-PAR in runControl.config).

13.5.2 The LDC event buffer size

The event buffer size in the LDCs is given a default value of 1024000 bytes. This value can only be changed by adding the following statements in the file ***${DATE_SITE_CONFIG}runControl.config***:

```
>TCLEVAL
set ldcEventBufferSize x
```

where *x* is the desired buffer size in bytes[1].

When the value is changed, it is necessary to destroy all the shared memory segments of the event buffers with IPCS (a simpler way is to bootstrap all the LDCs.).

13.5.3 Run-control configuration parameters

The parameters declared under the keywords RUNPAR, CNFPAR, LDCPAR and GDCPAR will appear in the run-control windows, either the main one or the one labeled Configuration parameters (Figure 13.1). The operator is supposed to set the values of these parameters in the corresponding entry fields on the windows.

- ***maxEvents*** - See description in Table 13.3.
- ***maxBytes*** - Maximum number of **kBytes** to be collected in a run. It is used to limit the amount of data to be handled by the offline analysis for a single run. The data may be saved on several files. according to the ***maxFileSize*** parameter. See also the description in Table 13.3.
- ***maxEventSize*** - It should indicate the maximum size in bytes of the sub-event in the LDCs. It is not used in the GDCs. The ReadEvent routine must make sure not to read more bytes than ***maxEventSize***, otherwise memory corruption may happen. On the other hand, ***maxEventSize*** cannot be indefinitely large, since the same amount of bytes will be wasted at the top of the event buffer (in any case it should be one order of magnitude smaller than the event buffer).
- ***phaseTimeoutLimit*** - See description in Table 13.3.
- ***randEventMinSize*** - Optional parameter. See description in Table 13.3.
- ***randEventMaxSize*** - Optional parameter. See description in Table 13.3.
- ***randEventInterval*** Optional parameter. See description in Table 13.3.
- ***maxFileSize*** - See description in Table 13.3.
- ***recordingDevice*** - To be set for all the LDCs. If not set, the LDC works in isolation and discards the events. It may indicate either a disk file, for local recording, or the name of the event builder GDC. In the latter case, it must terminate with ":".
- ***ebRecordingDevice*** - Event builder recording device.This indicates the directory and the name of the disk file where the full events data should be written.

  There is a special naming convention concerning the recording devices (both ***recordingDevice*** and ***ebRecordingDevice***) when they indicate a file name. At start of run, these names are communicated to the respective machines; before that, they are parsed and the following substitutions are applied:

  1. The first occurrence of the character "*@*" is replaced by the machine name.

1. The parameter ***recordingBufferSize*** defined in ***/date/runControl/rcShm.h*** cannot be changed by the operator. It is used to save the size of the event buffer *after* the shared memory segment has been created.

2. The first occurrence of the character "**#**" is replaced by the run number.

This feature allows us to generate data files with different names for each machine and each run. For example, if **ebRecordingDevice** is declared to be **Data.#.raw**, the data of the run 321 will be stored on the file **Data.321.raw**; the next run will be on **Data.322.raw**, and so on.

The actual communicated file name may be forced to be **/dev/null**, if the recording is disabled on the main run-control window.

If the name of the recording device ends by ":" it is supposed to indicate a remote machine instead of a file, therefore the substitutions do not occur.

- **monitorEnableFlag** - See description in Table 13.3.
- **logLevel** - It controls the generation of messages by all the date processes running on a DAQ machine. Standard value is **10**. All the possible values are the following:

  0 - No message is generated.

  10 - Normal error and information messages. No impact on the performance.

  20 - Detailed information messages. Small performance degradation.

  30 - Debugging information messages. Big performance degradation.

Figure 13.1 Run control configuration window

13.5.4 Run number

The run number is automatically incremented at each start of run. It cannot be easily changed by the operator.

The run control creates a file **${DATE_SITE_CONFIG}/runNumber.config** and stores there the last run number used by DATE. If you want to modify the current run number (in order to re-start the series) you may edit the file and replace the run number with the new number minus one.

An example of the file exists in the distribution: **/date/runControl/runNumber.config**, which may be copied in the right place and edited there. The file contains the line shown in listing Listing 13.4.

Listing 13.4 Repository of the run number

```
1:  set runNumber 0
```

Make sure that the file has got write access for everybody:

```
chmod a+w ${DATE_SITE_CONFIG}/runNumber.config
```

13.5.5 Multiple run controls

It is not allowed to have several run controls active at the same time. If more than one run control try to connect to the same machines, an alert window is opened, offering as only option "give up".

In the case of a crash of the run control while connected to the DAQ machines, it would be impossible to have a new run control connected to previously connected machines. This limitation can be overcome by doing the following operations:

1. Select in the main run control window the option View-Tcl eval.
2. In the **Tcl command** entry area, type the following command:

   ```
   set bypassMastership 1
   ```

3. Click on the **Eval** button. It is then possible to connect to the machines since the "take over" option is proposed as well.

This statement can also be permanently set in the file ***${DATE_SITE_CONFIG}/runControl.config***, under the **>TCLEVAL** keyword. Then it would be possible to have several run controls active at the same time. This practice is strongly discouraged.

13.6 Information logger configuration

The information logger makes use of a dæmon running on a host machine. The name of the machine must be indicated in the file ***${DATE_SITE_CONFIG}/infoLogger.config.***

This configuration file should contain one line as:

Listing 13.5 Repository of the host of the information logger dæmon

```
1:  DATE_INFOLOGGER_LOGHOST infologger_host
```

where *infologger_host* is the name of the machine where you want to run the infoLogger daemon.

13.7 Monitoring configuration

Functionally, hosts participating to a DATE monitoring scheme can be defined as:

1. **monitoring** hosts running specific **monitoring programs**, either written by a specific experience or part of the standard monitoring package (e.g., the utility **eventDump**);
2. **monitored** hosts offering **monitoring streams** to monitoring programs: these streams can be **online** streams (from a live DAQ system) or **offline** streams (typically files available from permanent data storage);
3. **relaying hosts** offering a liaison between **monitored** and **monitoring** hosts that cannot establish a direct link due to the presence of network firewalls or gateways.

It is possible to have any combination of those three functions, e.g. hosts who are monitoring, are monitored and offer relayed monitoring to other hosts.

The DATE monitoring scheme *needs* configuration only for **monitored** and **relaying** hosts in the following situations:

1. online **monitored** hosts (LDCs or GDCs) who wish to offer **online**, **offline** or **relayed** monitoring to itself and/or to other hosts;
2. hosts part of a DATE system who wish to offer **offline** or **relayed** monitoring to other hosts.

No setup is required for hosts only wishing to perform monitoring, either on the same or on remote hosts and a complete DATE installation is not required. For the developer of the monitoring program itself, a library is available and can be used in stand-alone mode. Otherwise, monitoring programs can be exported to any type of hosts (within the set of supported architectures) with no need for extra files or special setups. No daemons are necessary and no configuration is required on the monitoring hosts. If an upgrade from DATE V2 or earlier is performed, it is recommended (although not necessary) to remove all existing setups eventually present in the **monitoring** host.

We will now review the configuration needed on **monitored** and **relayed** hosts to let them perform their function.

13.7.1 Creation of configuration files

The monitoring scheme can be configured using three separate files:

- ***${DATE_SITE_CONFIG}/monitoring.config***: this file is optional and can be used to control a complete DATE site, all types of hosts;
- ***${DATE_SITE}/${DATE_HOSTNAME}/monitoring.config***: this file is mandatory for **online** hosts and *must* be created by the DATE system administrator. It is not required for **offline**, **relayed** or **monitoring** hosts;
- ***/etc/monitoring.config***: this file is optional and can be used to control the behaviour of **relaying** hosts; it is useless for **online** or **offline** monitored hosts.

The above files should be created using the following commands:

Listing 13.6 Creation of configuration files

```
1: > touch file
2: > chmod u=rw,g=rw,o=r file
```

where **file** is the full path of the file to be created. Once created, the configuration files can be edited and parameters can be specified as a list of names followed by their associated values. Comments can be inserted via the "#" sign, e.g.:

```
# This is a comment
PARAMETER    VALUE   # comment
```

These files can be changed at any time. Some of the parameters (those labelled in Table 13.4 as "Online monitoring only") require the acquisition to be stopped and no active clients (the command **monitorClients** - see - can be used to check for registered clients). All the other parameters can be changed at any time and will become active for all new clients (producers and consumers) started after the modification(s).

When the same parameter is defined in multiple files, a "last given" policy is followed, that is:

- parameters defined in *${DATE_SITE_CONFIG}/monitoring.config* can be overridden by equivalent definitions from any of the other files;
- parameters defined in *${DATE_SITE}/${DATE_HOSTNAME}/monitoring.config* are final for local monitoring and can be overridden by equivalent definitions from */etc/monitoring.config* for relayed monitoring;
- parameters defined in */etc/monitoring.config* are final and cannot be overridden.

Only exception to this scheme is the parameter *LOGLEVEL*, where the highest given level is used (e.g. if the values 0, 10 and 20 are specified, the used value will be 20).

The parameters that can be specified in the configuration files are:

Table 13.4 Monitoring configuration parameters

Parameter name	Used for	Description
LOGLEVEL	All types of monitoring	Level for error, information and debug statements generated by the monitoring scheme
MAX_CLIENTS	Online monitoring only	Maximum number of clients allowed to be registered simultaneously
MAX_EVENTS	Online monitoring only	Maximum number of events available for monitoring
EVENT_SIZE	Online monitoring only	Average event size

Table 13.4 Monitoring configuration parameters

Parameter name	Used for	Description
EVENT_BUFFER_SIZE	Online monitoring only	Size of buffer used to store events data
EVENTS_MAX_AGE	Online monitoring only	Maximum age (in seconds) of the events available for monitoring
MONITORING_HOSTS	Online monitoring Networked monitoring	Comma-separated list of hosts allowed to perform monitor-when-available from this host
MUST_MONITORING_HOSTS	Online monitoring Networked monitoring	Comma-separated list of hosts allowed to perform all types of monitoring from this host

For the MONITORING_HOSTS and MUST_MONITORING_HOSTS parameters, a comma-separated list of hosts should be given, e.g.

```
MONITORING_HOSTS      localhost, suxy, mppcxy05
```

In the above example, the hosts allowed to perform “normal” monitoring are the local host, all hosts whose name begins with *suxy* (*suxy01*, *suxy02* and so on) plus the host *mppcxy05*.

A host who is defined within the *MONITORING_HOSTS* list can only perform monitoring-when-available. To be able to perform 100% monitoring, a host must be in the *MUST_MONITORING_HOST* list.

If the parameter *MUST_MONITORING_HOSTS* is not specified, all hosts can perform 100% monitoring on the monitored machine. Conversely, if the parameter *MONITORING_HOSTS* is not specified, all hosts can perform monitoring functions on the given machine.

13.7.2 Installation of the monitoring daemon

All machines wishing to offer monitoring (online, offline or relayed) to other hosts should install the monitoring daemon mpDaemon. This can be done either via the dateNetwork command or manually (see 13.3.2 for more details concerning the installation of the daemon).

The proper setup of the daemon can be verified as follows. From any host wishing to perform monitoring, run the command

Listing 13.7 Remote monitoring test procedure

```
1:  > telnet host ${DATE_SOCKET_MON}
2:  Trying...
3:  Connected to host
4:  Escape character is '^]'.
5:
6:
7:  Connection closed.
```

Substitute “host” with the name of the machine offering monitoring functions. What reported above is a “good” example. When things go wrong, one should expect some error messages that should lead to the culprit of the problem, e.g.:

Listing 13.8 Example of failed remote monitoring installation

```
1:  > telnet host ${DATE_SOCKET_MON}
2:  Trying host...
3:  telnet: connect to address xxx.xxx.xxx.xxx: Connection refused
4:  telnet: Unable to connect to remote host: Connection refused
```

In the above example, the mpDaemon service has not been properly installed. Other useful debugging information may be available within the system console of the machine target (the “host” mentioned above). In our previous example, we can see the message:

host `inetd[4692]: execv /date/monitoring/SunOS/monServer.sh: No such file or directory`

A more complete validation of the remote monitoring scheme can be done using the eventDump utility facility part of the standard DATE monitoring scheme (for more details, see 4.3, “The “eventDump” utility program,” on page 47).

Hosts who wish only to perform outgoing monitoring functions do not need to install the monitoring daemon.