TECHNICAL NOTES

Framework for time-critical applications

User Support

Forecasts Department

February 2015

This paper has not been published and should be regarded as an Internal Report from ECMWF. Permission to quote from it should be obtained from the ECMWF.

European Centre for Medium-Range Weather Forecasts
Europäisches Zentrum für mittelfristige Wettervorhersage
Centre européen pour les prévisions météorologiques à moyen terme

European Centre for Medium-Range Weather Forecasts
Shinfield Park, Reading, RG2 9AX, United Kingdom

The information within this publication is given in good faith and considered to be true, but ECMWF accepts no liability for error, omission and for loss or damage arising from its use.

1 Introduction

This document defines the ”Framework for Member State time-critical applications” which was discussed during the 34th meeting of the TAC and approved by Council at its 61st session in 2004.

The document also provides the technical guidelines that must be followed by those wishing to make use of this framework. The technical guidelines are mainly meant for users wishing to set up activities using ”option 2” of the Framework.

If there are any aspects of this document that require further clarification or if there is any issue relevant to your work but not covered in this document, you are advised to contact your usual User Support contact point.

2 General information

2.1 Service options available

The ”Framework for Member State time-critical applications” comprises three options:

1. Option 1: Simple job submission monitored by ECMWF:
   - Suitable for single jobs or simple jobs with multiple steps.
   - Based on the ECaccess framework.
   - Service available to all registered users (no formal request needed).
   - See https://software.ecmwf.int/wiki/display/USS/Simple+time-critical+jobs for more information.
2. Option 2: Member State ecFlow (or SMS) suites monitored by ECMWF:
   - Suitable for more complex applications comprising several tasks with interdependencies between them.
   - The suites will be developed according to the technical guidelines described in this document.
   - To be requested by the TAC representative of the relevant Member State.
   - Monitored by ECMWF.
3. Option 3: Member State ecFlow suites managed by ECMWF:
   - Further enhancement of Option 2.
   - Requires an ecFlow suite, which has usually been developed under option 2 of the framework.
   - Application developed, tested and maintained by the Member State.
   - It must be possible to test the application using ECMWF pre-operational (e-suite) data.
   - Member State suite handed over to ECMWF.
   - Member State responsible for the migration of the application, e.g. when supercomputer changes.
   - Monitored by ECMWF.
   - ECMWF will provide first-level on-call support, while second-level support would be provided by the Member State.
   - To be requested by the TAC representative of the relevant Member State.

2.2 General characteristics of Member State time-critical work

The main characteristics of Member State time-critical work are:

1. The work needs to be executed reliably, according to a predetermined and agreed schedule.
2. It runs regularly, in most cases on a daily basis, but could be executed on a weekly, monthly or ad hoc basis.
3. It must have an official owner who is responsible for its development and maintenance.

2.3 Systems that can be used at ECMWF to execute time-critical work

Within this Framework, Member State users can use the general purpose server ”ecgate” and the High Performance Computing Facility (HPCF), provided they have access to the HPCF resources.

In general, users should minimise the number of systems they use. For example, they should use ”ecgate” only if they need to post-process data, which is not excessively computationally intensive. Similarly, they should only use the HPCF if they need to run computationally intensive work (e.g. a numerical model) and do not need to post-process their output graphically before it is transferred to their Member State.

Member State time-critical work may also need to use additional systems outside ECMWF after having done some processing at ECMWF: for example to run other models using data produced by their work at ECMWF. It is not the purpose of this document to provide guidelines on how to run work which does not make use of ECMWF computing systems.

2.4 How to request this service and information required

Every registered user of ECMWF computing systems is allowed to run work using ”option 1” of this Framework and no formal request is required. Note that the access to our realtime operational data is restricted. Users interested in running this kind of work should refer to the document entitled ”Simple time-critical jobs - ECaccess”. See http://software.ecmwf.int/wiki/display/USS/Simple+time-critical+jobs.

To run work using ”option 2” or ”option 3” you will need to submit an official request to the Director of Forecasting Department at ECMWF, signed by the TAC representative of your Member State.

Before submitting a request we advise you to discuss the time-critical work you intend to run at ECMWF with your User Support contact point.

Your official request will need to provide the following information:

1. Description of the main tasks.
2. The systems needed to run these tasks.
3. Technical characteristics of the main tasks running on HPCF: number of processors required, memory needed, CPU/elapsed time needed, size of the input and output files, software/library dependencies, system billing units (SBU) needed (if applicable).

4. A detailed description of the data flow, in particular describing which data are required before processing can start. This description must state any dependency on data that are not produced by any of the ECMWF models and include from where the data can be obtained and their availability time.

5. A proposed time schedule for the main tasks, stating in particular when it is desirable to have the result of this work available 'to the customers'.

ECMWF will consider your request and reply officially within three months taking into account, in particular, the resources required for the implementation of your request.

3 Technical guidelines for setting up time-critical work

3.1 Basic information

3.1.1 Option 1

ECMWF has enhanced the ECaccess framework to allow for scheduling and control of jobs at predefined events. The operational suite will "inform" the ECaccess system that a certain event has occurred and this will trigger the execution of all relevant Member State jobs. ECaccess will be fully in charge of the control of the jobs. Member State users and the operators at ECMWF have the possibility to monitor the work and take remedial actions when required using an interface part of the ECaccess framework. This service is described in:

`http://software.ecmwf.int/wiki/display/USS/Simple+time-critical+jobs`.

3.1.2 Option 2

As this work will be monitored by ECMWF staff (User Support during the development phase; the operators, once your work is fully implemented), the only practical option is to implement time-critical work using a suite under ecFlow (or SMS), ECMWF's monitoring and scheduling software packages. Given that SMS will gradually be phased out, we ask new developers of Option 2 activities to use ecFlow. We will therefore only refer to ecFlow in the remaining part of this document.

The suite must be developed according to the technical guidelines provided in this document.

General documentation, training course material, etc ... on ecFlow can be found at:

`http://software.ecmwf.int/wiki/display/ECFLOW/Home`.

No on call support will be provided by ECMWF staff but the ECMWF operators can contact the relevant Member State suite support person, if this is clearly requested in the suite man pages.

3.1.3 Option 3

In this case, the Member State ecFlow suite will be managed by ECMWF. The suite will usually be based on either a suite previously developed in the framework of "option 2" or on a similar suite already used to run

ECMWF operational work. The suite will be run using the ECMWF operational userid and will be managed by staff in the production section at ECMWF.

The suite will generally be developed following similar guidelines to ”option 2”.

The main technical differences are that ”option 3” will have higher batch scheduling priority than ”option 2” work and that the ECPDS system (ECmwf Product Dissemination System) will normally be used to transfer products obtained by ”option 3” work. With ”option 3”, your time-critical work will also benefit from first level on call support from the ECMWF Production Section staff.

3.2 Before implementing your ecFlow suite

You are advised to discuss the requirements of your work with User Support before you start any implementation.

You should test a first rough implementation under your normal Member State userid, using the file systems normally available to you and standard batch job classes/queues, etc ... following the technical guidelines given in this document.

You should proceed with the final implementation only after your official request has been agreed by ECMWF.

3.3 UID used to run the work

A specific UID will be created to run a particular suite under ”option 2”. This UID will be set up as an ”application identifier”: such UIDs start with a ”z”, followed by two or three characters.

No password will be assigned and access to the UID will be allowed using a strong authentication token (ActivIdentity token).

A person responsible will be nominated for every ”application identifier” UID. A limited number of other registered users can also be authorized to access this UID and a mechanism to allow such access under strict control will be available. The person associated with the UID and other authorised users have responsibility for all changes made to the files owned by the UID.

The UID will be registered with a specific ”policy” (”timecrit”) which allows access to restricted batch classes, restricted file systems.

3.4 General ecFlow suite guidelines

As mentioned earlier, ”option 2” work must be implemented, unless otherwise previously agreed, by developing an ecFlow suite.

The ecFlow environment is not set up by default for users on ecgate or on the HPC systems. Users will have to load the ecFlow environment with a module:

```
    module load ecflow
```

3.4.1 Port number and ecFlow server

The ecFlow port number for the suite has the format `"1000+<UID>"`, where `<UID>` is the numeric UID number of the userid used to run the work. The script to start the ecFlow server is available on ecgate and is called ”ecflow_start.sh”. A second ecFlow server can be started up for backup or development purposes. This second ecFLow server will be started with the ’-b’ option and will use the port number `"500+<UID>"`.

The syntax of the ecflow_start.sh command is:

```
Usage: /usr/local/apps/ecflow/4.0.6/bin/ecflow_start.sh [-b] [-d ecf_home
directory] [-f] [-h]
       -b        start ECF for backup server or e-suite
       -d <dir>  specify the ECF_HOME directory - default
/home/us/usl/ecflow_server
       -f        forces the ECF to be restarted
       -v        verbose mode
       -h        print this help page
       -p <num>  specify server port number(ECF_PORT number)  - default
1000+<UID> | 500+<UID> for backup server
```

Note that the port number allocation convention doesn’t guarantee that the two numbers associated with your UID are free. A port number may already be used by another user for ecFlow or it may be used by another application. If ’your’ default port number is not free, you will have to start the ecFlow server by specifying your own port number, using the option ’-p’. Authorised port numbers are between 1024 and 65536. We advise you to choose higher numbers.

The ecFlow server will run on ecgate and can be started at system boot time. Please ask User Support at ECMWF if you want us to start your ecFlow server at boot time. A cron job which regularly checks the presence of the ecFlow server process should also be implemented. The above script ecflow_start.sh can also be used to run this check under cron, e.g. like in:

```
5,20,35,50 * * * * ~/cronrun.ksh ecflow_start.sh 1>~/ecFlow_start.out 2>&1
```

with the script $HOME/cronrun.ksh containing:

```
#!/bin/ksh
export PATH=/usr/local/bin:$PATH
. ~/.profile
. ~/.kshrc
module load ecflow
$@
```

Depending on your activity with ecFlow, the ecFlow log file (~/ecflow_server/ecgb.*.log) will grow steadily. We recommend that you install either a cron job or an administration task in your suite to clean these ecFlow log files. This can be achieved with the command ecflow_client:

```
ecflow_client --port %ECF_PORT% --host %ECF_NODE% --log=new
```

3.4.2 Access to the job output files

We recommend the usage of the simple log server (Perl script) to access the output files of jobs running on the HPCF. This log server requires another port number, which will have the format "35000+<UID>", where <UID> is the numeric uid of the userid used to run the work.

The log server will run on the HPCF and can be started after system boot. The script /usr/local/bin/start_logserver should be used to start the log server on the HPCF. The syntax of the command start_logserver is:

```
Usage: /usr/local/bin/start_logserver [-d <dir>] [-m <map>] [-h]
       -d <dir>  specify the directory name where files will be served
                 from - default is $HOME
       -m <map>  give mapping between local directory and directory
                 where ecFlow server runs - default is <dir>:<dir>
       -h        print this help page
```

The mapping can consist of a succession of mappings. Each individual mapping will first give the directory name on the ecFlow server, followed by the directory name on the HPC system, like in the following example:

```
-m <dir_ecgate>:<dir1_hpc>:<dir_ecgate>:<dir2_hpc>
```

We recommend that you implement a cron job or define an administration task in your suite to check the presence of the log server process. The above script /usr/local/bin/start_logserver can be used for this purpose.

Note that the job output files of running jobs on HPC are kept on a local spool, which is not visble from the interactive nodes (cca and ccb). In order to see the job output files of running jobs, you will therefore need to start the logserver on cca-log and ccb-log. See further for more details.

3.4.3 Managing ecFlow tasks

EcFlow will manage your jobs. Three main actions on the ecFlow tasks are required: one to submit, one to check and one to kill a task. These three actions are respectively defined through the ecFlow variables ECF_JOB_CMD, ECF_KILL_CMD and ECF_STATUS_CMD. You can use any script to take these actions on your tasks. We recommend that you use the commands provided by ECMWF with the schedule module which is available on ecgate. To activate the module, you will run:

```
module load schedule
```

The command called ’schedule’ can then be used to submit, check or kill a task:

```
Usage: /usr/local/apps/schedule/1.4/bin/schedule <user> <host> [<requestid>]
<jobfile> <joboutput> [kill|status]

 Command used to schedule some tasks to sms or ecflow
         <user>:         %USER%
         <host>:         %REMOTE_HOST%, %SCHOST%, %WSHOST%
         <requestid>:    %ECF_RID% or %SMSRID% (only needed for [kill|status])
         <jobfile>:      %ECF_JOB% or %SMSJOB%
         <joboutput>:    %ECF_JOBOUT% or %SMSJOBOUT%
```

```
    By default /usr/local/apps/schedule/1.4/bin/schedule will submit a task.
```

An example is given in the sample suite in ~usx/time_critical/sample_suite.def.

Alternatively, you can use the commands task_submit or task_status and task_kill.

3.4.4 EcFlow access protection

EcFlow allows access to be restricted to one ecFlow server, using the ecf.list file in the $ECF_HOME directory. We recommend that you set up and use this file, mainly to allow ECMWF staff to monitor your suite and to prevent unintentional access by other users. A sample file is available in ~usx/time_critical/ecflow/ecf.list.

3.4.5 EcFlow suite design recommendations

Some key points to keep in mind when designing your suite:

1. The suite should easily run in a different configuration. It is therefore vital to allow for easy changes of configuration. Possible changes could include:

   - Running on a different HPCF system.
   - Running the main task on fewer or more CPUs, with fewer or more threads (if relevant).
   - Using a different file system.
   - Using a different data set, e.g. ECMWF e-suite or own e-suite.
   - Using a different ”model” version.
   - Using a different EcFlow server (while only ecgate is available to you, this is not relevant).
   - Using a different UID and different queues, e.g. for testing and development purposes.

   The worst that could happen is that you lose everything and need to restart from scratch. Although this is very unlikely, you should keep safe copies of your libraries, executables and other constant data files.

   To achieve flexibility in the configuration of your suite, we recommend that you have one core suite and define ecFlow variables for all those changes of configuration you want to cater for. See variable definitions in suite definition file ~usx/time_critical/sample_suite.def.

2. It is also important to clearly document the procedures for any changes to the configuration, if these may need to be run by, for example, the operators at ECMWF.

3. All tasks that are part of the critical path, i.e. that will produce the final ”products” to be used by you, have to run in the safest environment:

   - If possible, your time-critical tasks should run on the HPCF system. If this is impossible and your task runs on ecgate, be aware that this may block your time-critical activity, as currently there is no backup for this system.

- Your time-critical tasks should not use the Data Handling System (DHS), including ECFS and MARS. The data should be available online, on the HPCF (either in a private file system or in the MARS Fields Data Base (FDB). If some data must be stored in MARS or ECFS, do not make time-critical tasks dependent on these archive tasks, but keep them independent. See the sample ecFlow suite definition in ~usx/time_critical/sample_suite.def.
- Do not use cross-mounted file systems. Always use local file systems.
- To exchange data between remote systems, we recommend the use of rsync.

4. The manual pages should include specific and clear instructions for the operators at ECMWF. An example man page is available from ~usx/time_critical/suite/man_page. Man pages should include the following information:

   - A description of the task.
   - The dependencies on other tasks.
   - What to do in case of failure.
   - Whom to contact in case of failure, how and when.

5. The ecFlow functionality of ”late tasks” is useful to draw the ECMWF operators’ attention to possible problems in the running of your suite. Try to set the functionality for a few key tasks only, with appropriately selected warning thresholds. If the functionality is used too frequently or if an alarm is triggered every day, it is likely that no one will pay attention to it.

6. The suite should be self-cleaning. Disk management should be very strict and is your responsibility. All data no longer needed should be removed. The ecFlow jobs and job output files, if kept, should be stored (in ECFS), then removed from local disks.

7. Your suite definition will loop over many dates, e.g. to cover one year. Depending on the relation between your suite and the operational activity at ECMWF, you will trigger (start) your suite in one of the following ways:

   - If your suite depends on the ECMWF operational suite, you will set up a time-critical job under ECaccess (see option 1) which will simply set a first dummy task in your suite to complete. Alternatively, you could resume the suite, which would be reset to ”suspended” after completing a cycle. See sample job in ~usx/time_critical/suite/trigger_suite.cmd.
   - If your suite has no dependencies with the ECMWF operational activity, we suggest you to define a time in your suite definition file when to start the first task in your suite.
   - If your suite has no dependencies on the ECMWF operational activity, but has dependencies on external events, we suggest that you also define a time when to start the first task in your suite, and that you check for your external dependency in this first task.
   - The cycling from day to day will usually happen by defining a time when the last task in the suite will run. This last task should run sufficiently long in advance before the next run will start. Setting up this time will allow you to watch the previous run of the suite up until the last task has run. See the sample suite definition in ~usx/time_critical/sample_suite.def.

   Note that if one task of your suite remains in aborted status, this will NOT prevent the last task to run at the given time but your suite will not be able to cycle through to the next run, e.g. for the next day. Different options are available to you to overcome this problem. If the task that failed is not in the critical path, you can give instructions to the operators to set the aborted task to complete. Another option would be to build an administrative task that checks before each run that all tasks are set to complete, and therefore forces your suite to cycle through to the next run.

One key point in the successful communication between the jobs running on the HPCFs systems or ecgate and your ecFlow server is the error handling. We recommend the use of a trap, as illustrated in the sample suite in ~usx/time_critical/include/head.h. The shell script run by your batch job should also use the ”set -ue” options.

3.4.6 Sample suite

A sample suite illustrating the previous recommendation is available in ~usx/time_critical/sample_suite.def.

3.5 File systems

The UID used to run the work will be given a quota in $HOME (on the High Availability NFS server) and $SCRATCH on ecgate.

Ffile systems have been set-up on ecgate and on the HPCF clusters for time critcal applications: they are called /ms_crit on ecgate and /sc1/tcwork and /sc2/tcwork on the current HPC systems (cca and ccb). These file systems are quota controlled and therefore you will need to provide User Support with an estimate of the total size and number of files which you need to keep on this file system.

This file system should be used to hold both your binaries/libraries and input and output data.

No select/delete process will run on this file system and you will be required to regularly remove any unnecessary files as part of your suite.

You will also be required to safely backup copies of your binaries, etc into ECFS. It is recommended to include a task at the beginning of your suite, not to be run every day, that will restore your environment, in case of emergency (”restart from scratch” functionality).

If there is a need for a file system with different characteristics (e.g. to hold safely on line files for several days), these requirements can be discussed with User Support and a file system with the required functionalities can be made available.

3.6 Batch job classes/queues

A specific batch job queue has been set up on ecgate: it is called ”timecrit” and access is restricted to the UIDs authorised to run ”option 2” work only. This is the class/queue you should use to run any time-critical work on ecgate. If there are any non time-critical tasks in your suite (e.g. archiving tasks), these can use the other classes/queues normally available to users.

Similarly, on both HPCF clusters different specific batch job queues have been set up. They are called ”ts”, ”tf” and ”tp”, respectively for sequential work, fractional work (work using less than half of one node) and parallel work (work using more than 1 node). Again, you can use the other classes/queues for any non time-critical work.

When you develop or test a new version of your time-critical suite, we advise you to use the standard classes or queues available to all users. In this way, your time-critical activity will not be delayed by these testing or developments.

3.7 Data required by your work

Your work will normally require some ”input” data before processing can start. The following possibilities have been considered:

1. Your work requires input data which is produced by any of the ECMWF models. In such case it is possible to set up a specific dissemination stream which will send the required data to either ecgate or the HPCF depending on the requirements of your suite. ECPDS has also been enhanced to allow for the dissemination to a specific User ID (the UIDs used to run time-critical work) so that only this recipient User ID can see the data. With this enhanced system, the recipient User ID will also become responsible for the regular cleanup of the received data. This will make the ”local” dissemination option similar to the standard dissemination to remote sites. This is the option we would recommend.

   If produced by ECMWF, your required data will also be available in the FDB as soon as the relevant model has produced them and will remain online for a limited (variable depending on the model) amount of time. You can access these data using the usual ”mars” command. If your suite requires access to data which may no longer be contained in the FDB (e.g. EPS model level data from previous EPS runs) then your suite needs to access these data before they are removed from the FDB and temporarily store them in one of your disk storage areas.

   For no reason should any of your time-critical suite tasks depend on data only available from the Data Handling System (MARS archive or ECFS). Beware that the usage of the parameter ALL in any mars request will automatically redirect it to the MARS archive (DHS). Note also that we recommend you do not use abbreviations for a verb, parameter or value in your mars requests. If too short, these abbreviations may become ambiguous if a new verb, parameter or value name is added to the mars language.

2. Your work requires input data which is available at ECMWF but not produced by an ECMWF model. For example, your work requires observations normally available on the GTS e.g. if you are interested in running some assimilation work at ECMWF. In such a case you can obtain the required observations from /vol/msbackup/ on ecgate where they are stored by a regular extraction task running as part of the ECMWF operational suite.

   For any other data you may need for your time-critical activity and which is available at ECMWF, please contact User Support.

3. Your work requires input data which is neither produced by any of the ECMWF models nor available at ECMWF. You will then be responsible for setting up the required ”acquisition” tasks and establish their level of time criticality. For example, your suite may need some additional observations which improve the quality of your assimilation but your work can also run without them in case there is a delay/problem in their arrival at ECMWF. Please see the section ”Data transfers” for advice on how to transfer incoming data.

3.8 Data transfers

3.8.1 Outgoing data - sending data from ECMWF

We recommend the use of the ectrans command to send data to remote sites. The command has recently been enhanced to include a retrial of transfers from the spool.

We recommend set up the ectrans remote associations on your local ECaccess gateway. If this is not available, you can set up remote associations on the ECaccess gateway at ECMWF.

Note that, by default, ectrans transfers are asynchronous; the successful completion of the ectrans command does not mean your file has been transferred successfully. You may want to use the option ”-put” to request synchronous transfers.

More recently, we have enhanced the ECMWF dissemination system (ECPDS) to allow the dissemination of data produced by option 2 suites to remote sites. Please contact User Support for more information on this.

3.8.2 Incoming data - transferring data to ECMWF

We recommend ectrans (option -get) to upload some data from a remote site to ECMWF. Other options, including the possible use of ECPDS, may be considered in specific situations. Please discuss these with User Support.

3.8.3 Transferring data between systems at ECMWF

We recommend the use of scp/rsync to transfer data between ecgate and the HPC systems.

No use of NFS mounted file systems should be made to transfer data between the general purpose server ecgate and the HPCF systems.

We remind you not to use the DHS (MARS or ECFS) for any tasks in the critical path.

3.9 Scheduling of work

The UIDs authorised to run ”option 2” work will be given higher batch job scheduling priority than normal Member State work. All ”option 2” UIDs will be given the same priority.

The Centre’s core operational activities will always be given higher priority than MS time critical activities. If problems affecting the core activity suites arise or long system sessions are needed, the Member State suites and jobs may be delayed or possibly abandoned.

Member State users may need to consider the possibility of setting up suitable backup procedures for such eventualities.

3.10 Backup procedures

The UIDs authorised to run ”option 2” work have access to both HPCF clusters and are advised to implement their suite so they are ready to run on the cluster they normally do not use if the primary cluster is unavailable for an extended period of time.

The two separate HPCF environments (currently only the /sc1 and /sc2 file systems) should be kept regularly synchronised using utilities such as ”rsync”.

It should be possible to change the HPCF cluster used by the suite by doing a simple change of ecFlow variable (variable SCHOST in the sample suite). Similarly it is desirable to change the file system used by the suite by changing an ecFlow variable (variable STHOST in the sample suite).

Users may wish to consider the possibility of setting up more sophisticated backup procedures such as the regular creation of backup products based on previous runs of the suite.

3.11 Data archiving

Users wishing to set up Member State time-critical suites at the Centre should carefully consider their requirements regarding the long term storage of the products of their suite.

In particular, they should consider if they want to archive their time-critical application’s output data in MARS. In such a case, users are advised to contact their User Support contact point to start discussing the relevant technical issues which are beyond the scope of this document. The same recommendation applies to users wishing to consider the storage of their suite’s output in the on-line FDB (ECMWF’s Fileds Data Base). If your model/application producing the suite’s output is already enabled to store into the FDB, then the disk space to be used is /sc1/tcwork/ms_fdb or /sc2/tcwork/ms_fdb, e.g. FDB_ROOT=/sc1/tcwork/ms_fdb on each cluster.

For most users we recommend that their time-critical application’s output data is stored in the ECFS system, if this is required.

Please note that no time-critical task in your suite should depend on the completion of an archiving task.

Please also note that possible users of your suite’s products should be advised not to depend on the availability of such products in any part of the DHS system (both ECFS and MARS archive), as its services offered can be unavailable for several hours.

3.12 Making changes to the suite

Once your suite at option 2 is declared running in time-critical mode, we recommend you not to touch this suite any longer for new developments. We recommend that you define a similar suite in parallel to the time-critical one and that you first do the testing of changes under this suite. When you make important changes to your suite, we recommend that you inform your relevant User Support contact point and the ECMWF operators (newops@ecmwf.int).

At ECMWF, we will set up the appropriate information channels to keep you aware of changes that may affect your time-critical activity. The most appropriate tool is a mailing list.

3.13 Feedback

We welcome any comments on this document and the framework for time-critical applications. In particular, please do let us know if any of the above general purpose scripts does not fit with your requirements. We will then try to incorporate the changes needed.