

---

## Interface of access to Code\_Aster : ask

---

### Summary:

This document presents ask (contraction of *Aster* and *Tk*), the interface allowing to organize its calculations *Code\_Aster*.

One describes the logic of operation of ask, the graphic interface itself, bsf which makes it possible to explore the files on the various waiters, then one explains how the user can configure ask according to his needs.

For a rapid access with *Code\_Aster*, one shows starting from examples how to launch a study and how to make a parametric study.

For the people having access to the waiter of calculation *Code\_Aster* network EDF, one explains how to emit a card of anomaly, to reach the cards back from experiment (REX).

Lastly, in one second part, one presents *as\_run* which gives many opportunities on command line.

The features of ask version 1.1 are described here<sup>3</sup> who manages all the versions supported of *Code\_Aster*.

## Contents

1	General presentation and concepts.....	4
1.1	Starting.....	4
1.2	Features.....	4
1.3	Operating process.....	4
1.4	Waiters askt.....	5
1.5	Profile askt.....	6
2	Description of the graphic interface.....	7
2.1	Menu bar.....	7
2.1.1	Small File.....	8
2.1.2	Small Configuration.....	9
2.1.3	Small Tools.....	11
2.1.4	Small Options.....	12
2.1.5	Small Assistance.....	13
2.2	Mitres.....	14
2.2.1	Mitre STUDY.....	14
2.2.1.1	List of the types for STUDY.....	15
2.2.2	Button REX.....	16
2.3	Parameters of tender.....	17
2.4	Bar state.....	19
2.5	Follow-up of the jobs.....	19
3	Limp of selection of files: bsf.....	20
3.1	Navigation.....	21
3.2	Small File and finely contextual.....	22
3.3	Small Posting.....	22
4	Configuration.....	23
4.1	Waiters.....	24
4.2	Configurations.....	24
4.2.1	General preferences.....	24
4.2.2	Preferences Network.....	24
4.3	Tools.....	25
5	How to make a study?.....	26
5.1	Creation of the profile.....	26
5.2	Selection of the files.....	26
5.2.1	Definition of a basic way.....	26
5.2.2	Addition of existing files.....	26
5.2.3	Addition of files.....	26
5.2.3.1	... while inserting a blank line.....	27
5.2.3.2	... with "Value by default".....	27

5.2.4 To remove a file.....	28
5.3 Launching of calculation.....	28
5.4 Consultation of the results.....	29
5.5 Use of the tools.....	29
5.6 Advanced features.....	29
5.6.1 exectool.....	30
6 How to launch a parametric study.....	31
7 Launching of calculations in parallel.....	31
7.1 Distribution of calculations.....	31
7.1.1 Use of the available resources.....	31
7.1.2 Time of expiry.....	32
7.2 Activation of the parallelism of Code_Aster.....	32
7.3 Multiple executions.....	32
8 Use of as_run.....	33
8.1 For the user.....	33
8.2 For the developer.....	34
8.3 To maintain an installation local.....	34
8.4 For the interfaces launching of calculations.....	35
8.5 For the tasks of administration.....	35
8.6 To supervise the launching and the return of a calculation.....	36
8.6.1 Assumptions.....	36
8.6.2 Orders.....	36
8.6.2.1 Recovery of information of the waiter (permet also of to test connection).....	36
8.6.2.2 Launching of a calculation.....	36
8.6.2.3 Interrogation.....	37
8.6.2.4 Actualization of the state of the job.....	37
8.6.2.5 Recovery of the results.....	37
8.6.2.6 To remove the job.....	37
8.6.2.7 To consult the end of the output of a job in progress.....	38
8.6.2.8 To publish the file output.....	38
8.7 Options.....	38
8.7.1 Additional precise details.....	39
9 Frequently asked questions.....	40

## 1 General presentation and concepts

---

The provision of a tool like *Code\_Aster* who functions on multiple platforms requires a graphic interface to simplify the work of the user.

In addition, evolution of the centralized means of calculation and diffusion into free of *Code\_Aster* required the development of a portable product whose architecture can adapt to the various computer set-ups.

astk is the graphic interface which makes it possible to organize Lbe calculations *Aster* : to prepare Lbe given, to organize the files, to reach the tools the pre one and postprocessing, to launch and follow the evolution of calculations. astk also makes it possible to choose the version of *Code\_Aster* to use among those available (stable, testing...).

The name "astk" comes from the contraction from *Aster* and Tk which is the graphic bookstore used.

### 1.1 Starting

In Salomé-Meca, to open the module *Aster*, then to launch astk since its icon or since the menu *Aster/Tools/Run ASTK* .

On command line, it is necessary to carry out: `Salome Shell -- astk`, *Salome* being a script available in the installation of Salomé-Meca. One can find it while replacing `nnn` by the index of version in: `/opt/salome-meca/nnn/appli_Vnnn/salome`.

Apart from Salomé-Meca, the order `astk` has, in general, summer placed in the way by default of the users, the interface opens with a configuration by default (see [§ 4] *Configuration*).

If the order is not found, it is necessary to launch: `[aster_root] /bin/astk` where `[aster_root]` is the repertoire of installation of *Code\_Aster* (contact possibly your administrator).

### 1.2 Features

**STUDY** : astk makes it possible to launch a calculation *Aster* on the local machine (for example within the framework of a use on personal computer), on a waiter of departmental calculation or of course the waiter of calculation *Aster* EDF-R&D (access restricted to EDF and its people receiving benefits of studies).

**REX** : To make live the code passes by the taking into account of the remarks, needs, and opinions of the users. Cards back from experiment can be emitted and consulted by the users since the interface, the developers can answer it (access to necessary the EDF waiter). This functionality is reserved at EDF and to its people receiving benefits.

**MULTI-MACHINES** : Files necessary to these various actions (files of grid, source, results...) can be distributed on various machines on the network (declared in astk), astk ensuring the transfer and compression/decompression.

The passage of a firebreak is not proposed.

**TOOLS** : The user can launch various predefined tools and configure those which it needs (maillor, tool for postprocessing, editor...).

**BSF** : A browser is provided (called `bsf`, box of selection of files), it makes it possible to traverse the filesystems of the definite distant machines and to carry out current operations on the files: copy, suppression, name change, impression, or to open a window of orders on these machines.

**ASJOB** : One can follow calculations, their state (in particular in the case of launching in batch) since the window "Followed by the jobs", also called `asjob`.

### 1.3 Operating process

Architecture client/server authorizes a clear separation between the interface (customer) and the tools used to reach the code (services). The entrance point of the waiter is `as_run` : it makes it possible to use the version of reference in batch and/or to reach all the versions available on a network or in room.

Communication protocol by default between the various machines is HS for the Shell orders and SCP for the copy of files. One can however still use rsh and CCP but this is disadvised on most networks.

**Example:** The customer (i.e. the interface launched by the order `ask` or `codeaster-GUI`) request the launching of a calculation on a waiter of calculation.

The interface carries out a service `as_run` who undertakes to copy the data files from the waiter (in an intermediate repertoire defined by this one and shared by all the nodes of this waiter), to ask the waiter to start calculation. The results of calculation remain on the waiter of execution (in the intermediate repertoire if they are supposed being recopied towards the customer or in their final repertoire if their destination is on the waiter of calculation). The results are repatriated if necessary on the machine customer when state of calculation is ENDED in the follow-up of the jobs; i.e. after a manual or automatic actualization.

So only connection in the direction machine customer towards waiter is requested (key valid HS for a connection without password). There are no more the connections opposite initiated by the waiter towards the customer (as in the previous versions to the 1.9).

Moreover, once subjected calculation, the connection network between the customer and the waiter can be stopped without risk to lose calculation because the data were already deposited on the waiter and the results will be repatriated only on request of the customer (interface).

For the purists: `as_run` is not truly a waiter in the facts, it does not have there a demon with listening on a particular port. It is started with the request.

## 1.4 Waiters ask

For ask, a waiter is:

- that is to say a waiter of calculation *Aster* : a machine on which one can find `as_run`, i.e. the whole of the services which give access the code; one will be able to launch calculations via this waiter and to use files on this waiter for a calculation,
- that is to say a file server: one will be able simply to use files on this waiter during a calculation.

The waiter called "Room" is in fact a file server (only information of connection are necessary for a file server, but one easily knows them on the local machine).

If one wishes to launch calculations on the local machine (on which are installed a version of *Aster* and the waiter part of ask), it is also necessary to declare this machine like a waiter of calculation (Local and Machine in the following example and §2.1.2).

In this document, one will call "waiter of reference" the machine on which the version is managed qualified of *Code\_Aster* in-house EDF. Certain features are accessible only if the user has an access to this machine.

**Example of configuration:** (see [§ 2.1.2])

`ask` is used on the machine of name `mach00`, on which is installed *Code\_Aster*. One has access to a departmental file server `file01`, the files of this waiter are accessible (assembly NFS) since two machines from calculation `comp02`, `comp03`. One has also access to a cluster with NR nodes of calculation whose frontal machine is `front04`.

One has then:

- "Local" (reserved label): one has it always to explore the files which are on `mach00` ;
- "Machine" (label unspecified, the procedure of installation fixes this name at the value turned over by the order ``uname - N ``): whose address IP is that of `mach00` who is the waiter of calculation (services `as_run` installed) of the local machine;

- “Departure” (unspecified label): whose address IP is that of `file01`, which makes it possible to explore the files lodged by the file server, configured like a waiter of calculation Aster (services `as_run` installed) having `comp02` and `comp03` seen like nodes of calculation (`comp0i` can be the same machine as `file01`);
- “Cluster” (unspecified label): whose address IP is that of `front04` who is a waiter of calculation (services `as_run` installed) having NR nodes, the only accessible machine being `front04`.

The difference between “Departure” and “Cluster” is that in general the stations of calculation are accessible directly (one could thus subject a calculation in interactive on one or the other), whereas for a cluster, the users see only the frontal machine in general, the distribution being made on the nodes by a sequencer of batch processings.

One can as well have several waiters of calculation “Departure” or “Cluster” that not to have only the station “Local+Machine”.

## 1.5 Profile astk

A profile `astk` is a file which contains all the relative information with study: the site of the files of the study, in data, result, the type associated with each file, the parameters of tender of calculation (memory, time, machine of calculation, interactive batch/...).

The profile also contains parameters on the interface itself to resume a study exactly in the same state where it had been left.

The profile is recorded on request of the user (finely *File/To record* or *To record under...*) and with each launching of a calculation (it is the file whose extension is, by convention, `.astk`).

One will speak more far from the file “export” (extension `.export`) who is a simplified representation of the profile `astk`: it contains only information on the files and the parameters necessary to calculation (and no information on the graphic interface).

## 2 Description of the graphic interface

Graphic interface (IHM) breaks up into 4 parts:

1. A menu bar
2. Mitres
3. Parameters of tender
4. The bar of state
5. Zone of the arguments Aster

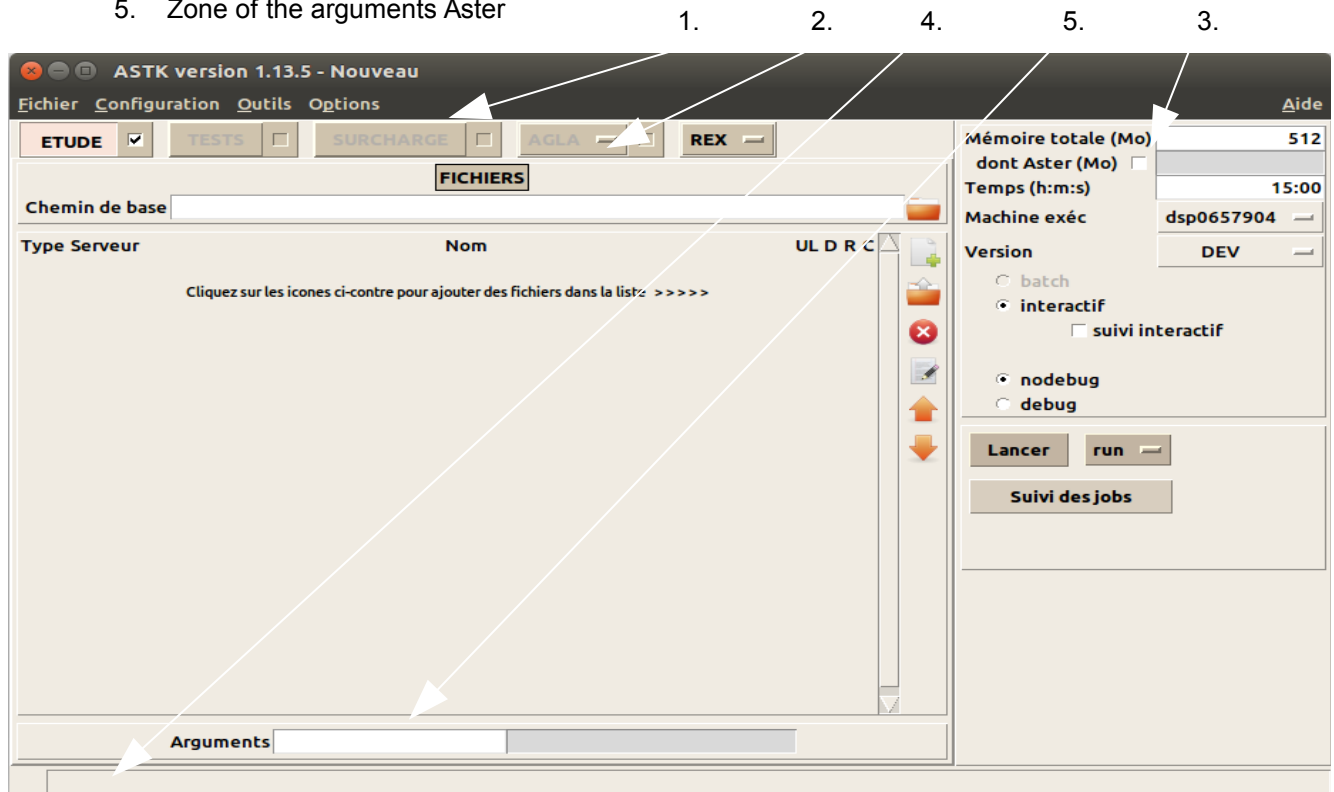


Figure 2-1: Principal window

The mitres TESTS, OVERLOAD and AGLA are now inactive.

### 2.1 Menu bar

The menu "Assistance" also makes it possible to reach the description of the menus.

## 2.1.1 Small File

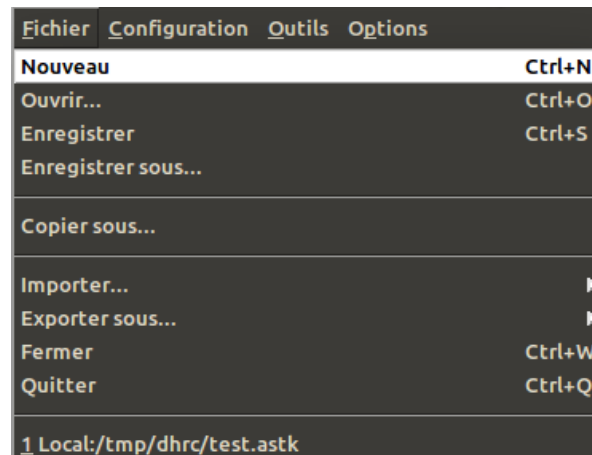


Figure 2.1.1-1: Small File

- **New** : Allows to create a new profile.
- **To open** : Open a profile created before by `astk`.
- **To record** : Safeguard profile running (this is done automatically during the launching of a calculation).
- **To record under** : Safeguard profile running while giving the choice of the name of the profile and its site.
- **To copy under** : Copy the profile running in another repertoire and proposes to copy the files (in data and/or results) that it reference.
- **To import** :
  - `.export astk` : A file reads again `.export` of a preceding execution (that can be the file `*.pNNN repertoire $HOME/flashor`).
  - `.export astk (addition mode)` : contrary to the classical importation which starts from a virgin profile, the files and repertoires present in the file `.export` are added to the current profile. The parameters and arguments are ignored. That makes it possible for example to import a study in a profile of overload.
  - `.export` of a card REX : Import the files attached to a card of the REX which one gives the number.
  - `CAS-test` : Import the files necessary to the launching of a CAS-test. The files are taken on the selected object computer.
  - `CAS-test (addition mode)` : idem + the files are added to those already present in the profile (including the overload).
- **To export under** : Allows to produce the file `.export` current profile.
- **To close** : To close the current profile
- **To leave** : Finish the application
- The NR the last open profiles are directly accessible starting from the small File.



## 2.1.2 Small Configuration

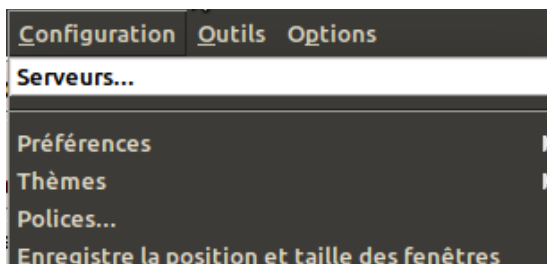


Figure 2.1.2-1: Small Configuration

- **Waiters** : Allows to modify the configuration of the waiters: waiters of calculation or file servers (see Figure 2.1.2-2).  
For the waiters of calculation, one chooses the mode of download of the configuration *Aster* : the waiter turns over the list of the versions available, the procedure (batch and/or interactive) and the limits associated.  
It is necessary to indicate the 'login' to use to connect itself to the waiter, and the repertoire where is installed *Code\_Aster*.  
For the file servers, to choose "any" like mode download the configuration *Aster*.
- **Preferences** : Defines the preferences of the user:
  - **General** (See Figure 2.1.2-3)  
For *astk* :  
Name, first name, e-mail will be used for the link towards the management tools of the experience feedback (emission, consultation of the cards of anomalies, evolution...).  
The access path to the editor and the terminal (window *xterm*) on the local machine, the version to select by default, the number of remanent profiles in the small File, the level of message (debug).  
One can choose to display all the types of known files, a short list or a choice staff, sorted or not by name (see §2.2.1).  
For the follow-up of the jobs:  
The number of lines displayed during the consultation of a calculation in the course of execution (tail), and the automatic frequency of actualization in minutes.
  - **Networks** (See Figure 2.1.2-4)  
Domain name network of the machine and if mode DHCP/VPN (addresses dynamic IP) is active, possibilities of fixing the variable *DISPLAY* for the posting of the external applications, the communication protocols used for the orders Shell (*rsh* or *HS*) and the copy of files (*CCP*, *SCP* or *rsync*).  
If one works only locally or if all the machines are in the same network field, one can leave empty domain name and be unaware of the warning with starting.  
**Caution** : to use the option *DISPLAY* with full knowledge of the facts and only if the value by default is not appropriate. To leave the field empties to let *astk* determine only the *DISPLAY* (according to its value with the launching of *astk*, of domain name...).
- **Colors** : Classic, KDE3 Hook, Nostalgic, Personalized,...
- **Police**
- **Record** the position and cuts windows

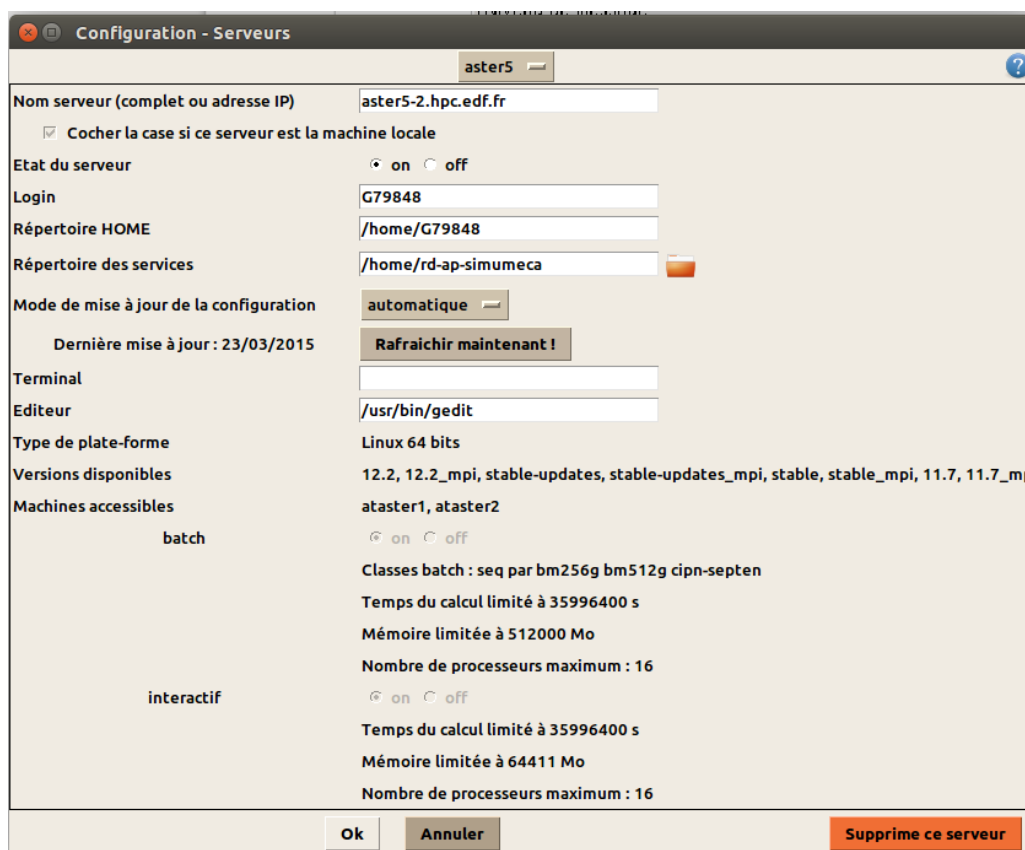


Figure 2.1.2-2: Small Configuration/Waiters

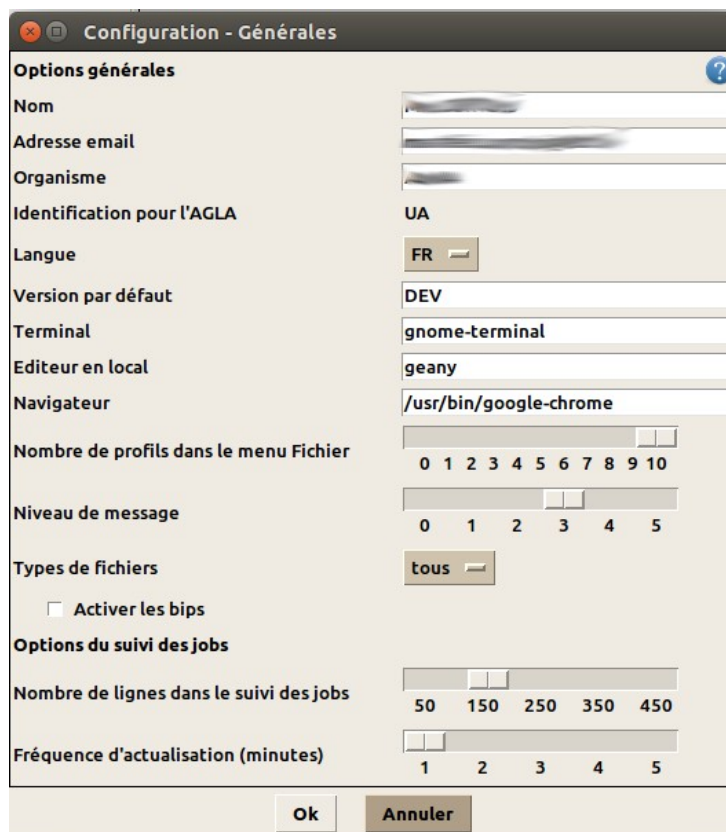


Figure 2.1.2-3: Small Configuration/General Preferences/



Figure 2.1.2-4: Small Configuration/Preferences/Network

**Remarks**

That one chooses protocol rsh/CCP or HS/SCP, connections must function without password: files .rhosts correctly filled or private/public keys valid SSH.

According to what turns over the order "uname" on the machine, it happens that astk thinks of being on a distant machine and then makes a rsh or HS to carry out a local order... In this case, the commonplace connection of the machine towards itself must also be configured.

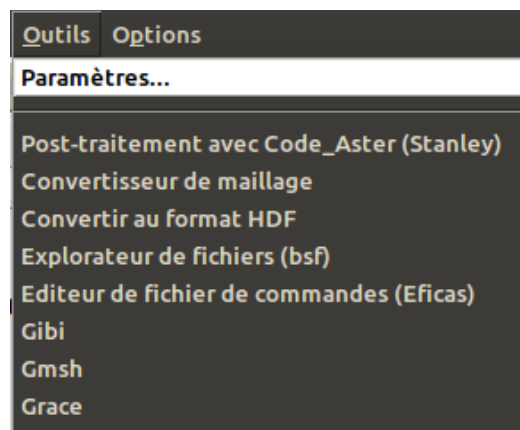
**2.1.3 Small Tools**

Figure 2.1.3-1: Small Tools

- **Parameters** : Allows to configure the launching of the tools.  
Certain tools are envisaged out of standard (without inevitably being available on the local machine):
  - Postprocessing with Code\_Aster : Tool of postprocessing, order STANLEY,
  - Converter of grid: Allows to convert a grid of a format towards another, the formats available are: aster, med, gmsh, ideas and gibi,
  - To convert with format HDF: A classical base Aster with format HDF makes it possible to convert,
  - bsf : Explorer of files multi-machines,
  - Eficas : Editor of command file Aster,
  - Gibi : Maillor and tool for postprocessing (usable free with Aster),
  - Gmsh : Maillor and tool for postprocessing (free),
  - grace : Graph plotter 2D.

When one chooses throw Stanley, astk preferably seeks in the profile the bases available (by order one base as a result, if there is not, one bhdf (HDF bases) as a result, if not one base in data and finally, one bhdf in data), a temporary profile starting from the profile produces running

with a command file Aster which starts with CONTINUATION ( ) and which launches STANLEY ( ).

The user can add his own tools, to parameterize the access path to the tools (including modifying the ordering of access to the standard tools), to define in which types of files associate the tool and to specify if one can use the tool on a distant file.

The following codes can be used in the command line:

- @F: absolute way of the selected file,
- @R: repertoire containing the selected file,
- @f: name of the file (without the repertoire),
- @D: address of the DISPLAY (that known at the time of the launching of the interface).

The tools are called either starting from small the Tools, or starting from the contextual menu on a file of a list or in the explorer (right button).

To select a file, it is enough to click on its name in a list (in the mitre Study, Tests or Overload).

While double-clicking on a file, association between the type of the file and the tool to be used are based on the extension of the file name for the bsf, whereas the type selected by the drop-down list (cf description of the mitres) prevails in astk.

## 2.1.4 Small Options

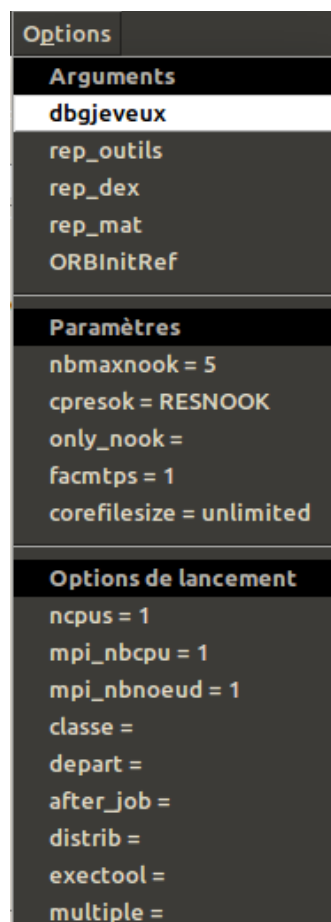


Figure 2.1.4-1: Small Options

- Arguments:  
Allows to position optional arguments which will have passed on the command line of the execution of Code\_Aster.

dbgjeveux : activate a mode different from the management of the objects in memory to detect crushings and destruction of objects,  
rep\_outils : the repertoire defines where the tools called are sought since Aster (onmetis or gibi for example),  
rep\_dex : the repertoire defines where are sought the external data (given grid),  
rep\_mat : the repertoire defines where the data of the catalogue material are stored.  
ORBInitRef : allows to contact a session active Salomé (this value is automatically positioned when astk is launched since Salomé).

The selected arguments appear in the grayed part of the zone of the arguments passed to Code\_Aster. The left part of this zone is free.

- Parameters

It is a question of defining optional parameters which will be written in the file .export. Three first are used during the launching of tests.

nbmaxnook : maximum number of invalid CAS-tests (NOOK, ARRET\_ANORMAL...) before the interruption of the list of the tests,

cpresok : allows to choose which are the tests which one keeps the files result,

RESOK: one keeps the files of tests OK and NOOK,

RESNOOK: one keeps only the files of tests NOOK,

facmtps : multiplicative factor of the time of the tests (compared to the time of reference of the para). Useful, for example, when one subjects tests with achievable built in debug mode, slower.

corefilesize: size limits file core at the time of a planting

- Options of launching

ncpus: the number of processors used by solvor MULT\_FRONT defines

mpi\_nbcpu: the number of processors for parallelism MPI defines

mpi\_nbnoeud: the number of nodes for parallelism defines MPI (Where them mpi\_nbcpu processors will be distributed)

class: allows to choose the class batch (or groups it of class) in which calculation will be subjected. It should obviously be checked that the class exists and that the parameters time and memory are compatible with this class.

departure: allows to differ the departure from a calculation. The time of departure is provided to the format hour: minute (to specify the starting day, syntax depends on the manager of work).

distribution : launch a parametric study (value yes/not, to see §6).

exectool : launching by using a particular tool (see §5.6.1)

multiple : carry out the profile on several machines (value yes/not, to see §7).

## 2.1.5 Small Assistance

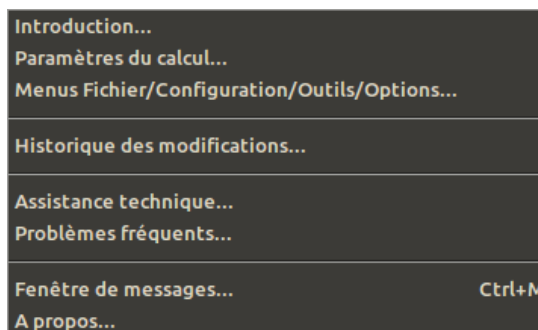


Figure 2.1.5-1: Small Assistance

- Introduction/Small Parameters of calculation/: access to the text of assistance
- Parameters of calculations: allows to define the size memory, the version of Code\_Aster who will be used, the way of calculating (Batch or interactif),...
- Menu File/Configuration/tools/options...

- History of the modifications: Evolution of askt to the wire of the versions, new possibilities, corrections carried out, known anomalies...
- Technical assistance...
- Frequent problems: Some questions frequently asked with their solutions.
- Window of the messages: The messages of information <INFO>, errors <ERREUR> are written in this window. With a level of message equal to or higher than 1 (see [§ 2.1.2]), information more or less is obtained.
- By the way...: the impossible to circumvent window of information.

## 2.2 Mitres

One finds five buttons in this zone. One speaks about mitre when the contents which one finds under the button change according to that which is in a hurry. Graphically, they are not "true" mitres because the widget does not exist in standard the Tk bookstore (and one did not want to add pre-necessary additional!).


STUDY is a mitre in which one informs the list of the files necessary to launch a study. REX is one simple button who displays a list of choice. The other buttons are now disabled.

### 2.2.1 Mitre STUDY

To reach the files more easily, to traverse the tree structures more quickly (files being often gathered in close repertoires), or to simplify the posting of the file names, one can define one **chemin de bases**.



Figure 2.2.1-1: Basic way

One defines it while clicking on the button To traverse  and one chooses the repertoire which will be the repertoire by default.

Each mitre contains a list of files (two lists for OVERLOAD).

Type	Serveur	Nom	UL	D	R	C
comm	Local	./forma00.comm	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
mmed	Local	./forma00.mmed	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 2.2.1-2: List of files and repertoires

One defines a file or repertoire by line. One finds left towards the line:

- Type: type of the file or the repertoire;
- Waiter: name of the waiter on which is the file/repertoire;
- Name: access path to the file: in absolute if it starts with '/', into relative compared to the basic way in the contrary case;
- UL: logical number of unit associated with this file;
- D, R, C: check boxes to indicate if the file/repertoire is respectively in data, result, compressed (with gzip).

When one changes the type of a file/repertoire, values by default are positioned for the indicators D, R, C and the logical number of unit.







#### Notice on the types:

*Code\_Aster handles the files via the procedures FORTRAN which use numbers of logical unit (file `fort.19` for example); it thus affects numbers of logical unit by default to simplify the life of the user. The "type" allows thus to easily reach the numbers used*



by default; it as makes it possible astk to check as the user provides coherent data (for example which one provides well a repertoire for such type.

To handle the list, one has six buttons:

-  : addition a new virgin entry at the end of the list;
-  : addition a file/repertoire at the end of the list by traversing the tree structure of files;
-  : remove the line currently selected in the list. A click-right makes it possible to remove all the lines;
-  : open the file/repertoire currently selected in the list with the editor of the waiter where the file (cf [§ is 4.1]). If it is about a repertoire, all the files of the repertoire are open with the editor (attention with the repertoires containing many files or binary files not éditables!);
-  : move a line upwards;
-  : move a line downwards.


## Arguments :

This zone of text makes it possible to transmit arguments to the achievable one *Aster*. See also [§2.1.4].

One can for example launch calculation in interactive and to require so that invites it orders is proposed at the end of the instructions to manually continue the execution of orders with the option “`--interact`”.

## Finely contextual:

While clicking with the right button on an entry of the list, one reaches a contextual menu:

- To open: launch the tool associated with this kind of file (determined by the “Standard” field, and not the extension of the file), if no tool is associated with this type, one publishes the file;
- To publish: publish the file (or all the files of the repertoire) in the same way as the button  ;
- To open with...: one can choose to open the file selected with one of the tools available (the result can be astonishing if the tool does not know this kind of file!);
- Values by default: the interface determines a file name by default according to the “Type” chosen starting from the name of the profile (file `.astk`), no value is proposed if the profile were not recorded yet (is used in general on a virgin line which one has just inserted, indicators D/R/C are not affected by this operation);
- Terminal: allows to open a final window `xterm`
- Properties: poster permissions, size, the date and the hour of the file (order `ls -lt`).

### 2.2.1.1 List of the types for STUDY

COM:	command files <i>Aster</i> (including the files of continuation)
e-mail:	file grid with the format <i>Aster</i>
wander:	file of error (fort.9 of <i>Aster</i> )
mess:	file of the messages of the execution
resu:	file of result (impression of the tests, impression to the format <i>Aster</i> )
base:	repertoire containing the base of calculation
bhdf	repertoire containing the base of calculation to format HDF
cast:	file result with format CASTEM
mast:	kept for reason of compatibility
mgib:	grid with the Gibi format
mmed:	grid with format MED
msh:	grid with the Gmsh format
msup:	grid with format IDEAS
datg:	command file Gibi

pos:	file result with the Gmsh format
ensi:	repertoire result with the Enight format
dat:	file result containing of the curves to format XMGRACE
PS:	file with the format postscript
agraf:	file result containing the data for Agraf (the old versions of Aster wrote the directives and the data in the same file that it was necessary to cut out with the order <code>post_agraf</code> on the machine of reference)
digr:	file result containing the directives for Agraf
rmed:	file result with format MED
unv:	file result with the "Universal" format (IDEAS)
distr:	file of the values used for a parametric study
hostfile:	file describing the resources machines to be used (parametric study)
name:	to recover starting from its name, a file present in the temporary repertoire /tmp
para:	file of parameters (retranscription of the parameters of calculation for the tests)
repe:	result and/or data directory (makes it possible to transmit/recover the complete contents of a repertoire; as one does not affect numbers of logical unit <i>Aster</i> must reach the files by their names).
	repertoire of the results at the time of a parametric study
libr:	file or repertoire with the choice of the user
btc:	script of launching generated by the service (one can thus recover it, to modify it...).

At the time of a astout on the machine of reference, `resu_test` must be on this one.

At the time of a parametric study, the repertoire of the results (standard `repe`) must be on the object computer.

## 2.2.2 Button REX

This button allows:

- **OfTo emit a card without joining the profile** : This function makes it possible to the user to emit a card back from experiment to point out an anomaly in *Code\_Aster* (AL: anomaly software), to ask an evolution of *Code\_Aster* (EL: evolution software), in one of the associated tools (Mongrel, Lobster, Eficas, astk, bsf...) (AO: anomaly outil/EO: evolution tool), a modification of documentation (ED: evolution documentation), or a request for expert testimony in modeling (AOM: assistance with the optimization of modeling). The relative information with the transmitter of the card are accessible by the menu *Configuration/Interface*. The files contained in the profile running are not joined to the card, which can make very difficult the treatment of the card. It is advised to use "to emit a card and to associate the files with it". The version of *Code\_Aster* indicated is that selected in the interface (*Parameters of calculation*).
- **OfTo emit a card and to associate the files with it** : This function makes it possible to emit the same type of card as the preceding one, this time the data included in the profile are united with the card. For an anomaly, the files to reproduce the error should always be provided. What does not prevent the user from trying to insulate as much as possible the encountered problem, in particular to try to join a study on a model which requires little memory and computing time!
- **Of To consult the cards** : gives access the management tools back from experiment (on the machine of reference).

### Notice

| *Features REX are available only on the waiter of reference.*



The screenshot shows a window titled 'REX' with a form for emitting an anomaly card. The form is divided into two main sections: 'Emetteur' and 'Fiche'.  
**Emetteur**  
Nom\* : M.UTILISATEUR  
Adresse email\* : username@domain.org  
Organisme\* : Entreprise ABCD  
(\*) Vous pouvez modifier ces valeurs dans le menu Configuration>Interface.  
**Fiche**  
Type : Choisir une catégorie (dropdown menu)  
Titre de la fiche : (text input field)  
Version : STA10  
Fichiers associés : Non  
At the bottom of the window, there are two buttons: 'Envoyer' and 'Annuler'.

Figure 2.2.2-1: Emission of a card of anomaly

## 2.3 Parameters of tender

The parameters of calculation are provided in the right part of the principal window.

The screenshot shows a dialog box for calculation parameters. It contains the following fields and options:  
Mémoire totale (Mo) : 512  
Temps (h:m:s) : 15:00  
Machine exéc : localhost (dropdown menu)  
Version : DEV (dropdown menu)  
Radio buttons for execution mode:  
 batch  
 interactif  
     suivi interactif  
Radio buttons for debugging:  
 nodebug  
 debug  
At the bottom, there are three buttons: 'Lancer', 'run' (dropdown menu), and 'Suivi des jobs'.

Figure 2.3-1: Zones of the parameters of calculation

One defines the quantity of:

- Total memory used for the job (in megabytes).
- the maximum time of calculation (in seconds, minutes: seconds or hour: time: seconds).

One chooses on which machine calculation is carried out, the version of *Code\_Aster* used, if calculation is subjected in batch or interactive. The option "followed interactive" makes it possible to carry out calculation while following its execution in a window of the final type. If calculation is launched on a distant waiter, this terminal is carried out remotely. Therefore the interface is then blocked as much as the execution of calculation is not finished to preserve open connection HS.

debug/nodebug: for a study without overload, one specifies which achievable one wishes to use (provided both are available).

The button "Throw" carries out the actions according to the notched mitres.

Followed button the "by the jobs" opens the window described afterwards.

### **Operating process** (for a STUDY):

During the launching of a study, a button of option is available beside the button "Throw". Three modes of launching are available:

- "run": carry out the study (classical operation),
- "dbg": launch the study by using the débogueur,
- "pre": prepare the repertoire of work without carrying out the study.

When one selects "dbg" or "pre", the mode "debug" is selected by default.

## 2.4 Bar state

The zone of text located all in bottom of the principal window provides of the assistance when one surfs in the menus, or when the pointer passes to the top of the buttons of the interface. During the launching of a calculation, the operations in progress are displayed here.



Groupe de classe batch (laisser vide pour utiliser les classes par défaut)  
Valeurs possibles : voir Configuration/Serveurs.

Figure 2.4-1: Bar state

## 2.5 Follow-up of the jobs

This window provides information on the state of launched calculations, which appear in the form of a list. While clicking with the right button on a calculation, one has the following choices:

- **To publish** breaks up into "File output", "File error" and "Files export" which gives access to the messages sent by the job on the standard exit (`stdout`) and the exit of error (`stderr`). One can also reach the file `.export` used during the execution. A double-click on a job of the list also displays the file output. These files are initially recopied locally in a temporary repertoire then open with the local editor.
- **To download the files results** force the recopy of the files results. This action is made automatically when calculation passes in state ENDED. If this stage failed (problem network), that would make it possible to carry out it again.
- **To bring up to date** question the waiters on which calculations are in progress. The check box makes it possible to reactualize automatically at a frequency defined in *Configuration/Interface*.
- **To bring up to date all** the state of all not finished calculations refreshes.
- **To remove** erase the selected jobs of the list, the files related to this job in the flashor and stops calculation if this one is not finished (by sending signal SIGKILL).
- **To stop properly** send signal SIGUSR1 to calculation. *Code\_Aster* intercept this signal to stop calculation, then the firm properly base which will be repatriated (if a base is as a result in the profile).
- **To purge flashor** traverses all the waiters and removes there the files of the repertoire flashor nonaccessible since the follow-up from the jobs.
- **To search** allows to consult the last lines of the file message of a job *in the course of execution* (it does not do anything on a finished job). One can use the zone of text "Filters" to display only the lines containing the character string indicated (in the form of regular expression).

The zone of text makes it possible to follow the advance of the requests carried out on the distant waiters.

Each line corresponds to a job, one finds 13 columns:

- The number of the job (in batch), number of the process in interactive
- The name of the job (name of the profile for a study, an overload, or name of function AGLA)
- Date of tender
- Hour of tender
- State of the job (HANGS, RUN, SUSPENDED, ENDED)
- Diagnosis of job (OK, NOOK, <A>\_ALARM, <F>\_ERROR, <F>\_ABNORMAL...)
- Name of the tail in batch or "interactive"
- Time CPU of the execution *Aster*
- Login on the waiter of calculation used
- Address of the waiter of calculation used
- Machine of calculation (name of the node for a cluster)
- Version of astk
- Indicating interactive batch/

## 3 Limp of selection of files: bsf

bsf is a tool delivered with askt which can be launched only. It is about an explorer of files who allows to surf on the local machine, like a classical explorer of files, and also on the various configured distant waiters.

bsf uses the configuration of the waiters of askt, in particular the fields necessary to connection (addresses IP, login) and the orders to open a terminal or an editor.

The reading of the configuration is made only with the starting of bsf, if one modifies the configuration in askt, it is thus necessary to close the bsf then to open it again.

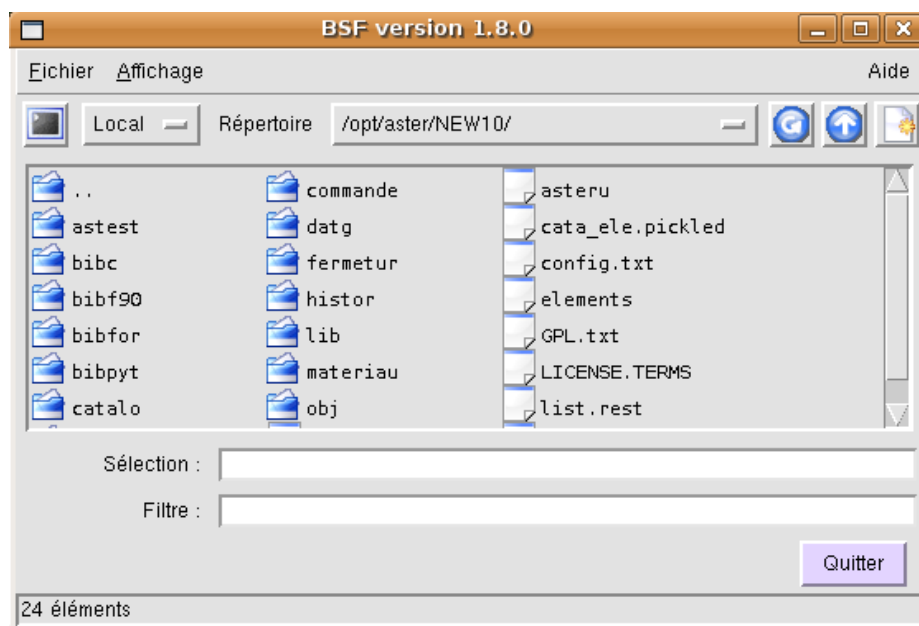


Figure 3-1: bsf

### Note:

*bsf does not treat the repertoire and file names containing spaces (it results from it a shift in the names and the types from the following objects).  
bsf does not have much any more of interest, the managers of standard files under Gnome or KDE replace it advantageously.*





## 3.1 Navigation

One finds two drop-down lists in the window of the bsf.

The first makes it possible to pass from a waiter to another, the second memorizes the list of the last twelve repertoires where the user carried out an action (edition, copy...). The first entry of this list is '---- Direct ----', which makes it possible to go directly in a repertoire which one seizes the name.

Generally, when the required repertoire does not exist, one turns over in the HOME defined for the current waiter.

One can configure this list and fix certain repertoires (so that they remain present in the list) while clicking with the right button on this one (cf Appears 3.1- has).

-  : the contents of the current directory refresh;
-  : go back to the repertoire relative;
-  : propose to create a new repertoire in the current directory (and places itself in this new repertoire);
-  : open a terminal on the current waiter.

The bar of state gives indications on the significance of these buttons when the pointer of the mouse passes to the top.

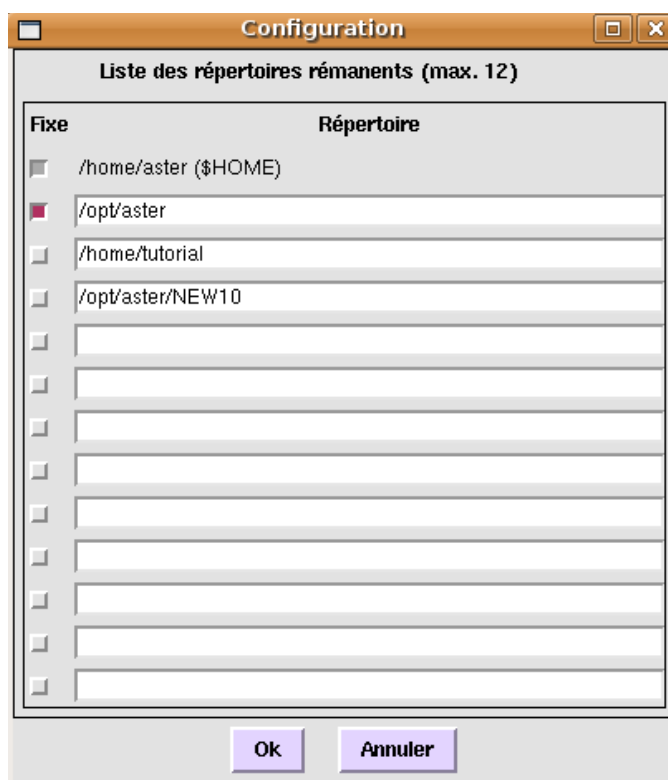


Figure 3.1-1: Window of configuration of the remanent repertoires

## 3.2 Small File and finely contextual

One finds certain entries similar to the contextual menu of the files in the mitres (see [§2.2.1]): To open, Publish, Open with..., Properties.

- **To copy** : allows to copy one or more files/repertoires (shortened by CTRL+C).
- **To cross** : idem To copy, except that the original files will be removed (shortened by CTRL-X).
- **To stick** : carry out indeed the copy or the displacement of the files (shortened by CTRL+V).
- **To remove** : erase the selected files/repertoires.
- **To re-elect** : give a new name to a file
- **New New file/ repertoire** : respectively allow to create a file or a repertoire in the current directory.
- **To carry out one order** : give the possibility of carrying out an order in the current directory, the current selection is proposed on the command line, it is necessary to choose one of the shells available on the waiter.
- **Terminal** : open one terminal on the current waiter.

## 3.3 Small Posting

One can choose to display or not the files known as hidden (starting with '.') by notching the box envisaged.

bsf poster the repertoires then the files, one alphabetically can Trier or according to the date in the normal or opposite order by modifying the options of the menu.

The preferences of posting are preserved if one selects **To record the preferences**.

## 4 Configuration

---

The first time that the user launches the interface, it has a configuration by default which was defined at the time of the installation. The configuration is then stored in the repertoire `$HOME/.astkrc`. If the user wants to return to the configuration of origin, it must erase this repertoire and start again `astk`.

### Notice n°1

*With each time one defines an order to be carried out (final, editor...), it is advised to show the absolute way (since the root) to prevent that the order is not found if the variable `$PATH` is incorrect.*

### Notice n°2

*When `astk` carries out an order which tries to display windows: if the order is carried out locally, not problem. If it is carried out on a distant machine, it is by using it `HS - X`, therefore posting must occur well if waiter `HS` authorizes it.*

The `as_run` part contains tools simplifying certain tasks of the developers like the consultation of the source code or the update of a local version of development.

The file `$HOME/.astkrc/prefs` thus contains two necessary information to contact the waiter of reference: the complete name of this waiter and the login of the user. This last must be defined in this file to avoid following alarm:

```
<A>_ALARM          remote connection may fail:
devel_server_user not defined in /home/xxxxxx/.astkrc/prefs
```

## 4.1 Waiters

One reaches the window of configuration by the menu *Configuration/Waiters* (see Figure 2.1.2-2).

The first button makes it possible to pass from a waiter to another and to add a "New waiter".

The fields are:

- Complete name or addresses IP: it is the name of the waiter on the network; one can indicate his complete name with domain name (for example: linux.labo.univ.fr) or its address IP (for example: 156.98.254.36).
- The box indicates that astk should not use rsh or HS to contact this machine.
- State of the waiter: one can put on "off" a temporarily inaccessible waiter.
- Login: identifying with which one connects oneself to the waiter.
- Repertoire HOME: repertoire by default when one surfs on this machine.
- Repertoire of the services: repertoire where are installed the services on this waiter (to show the way of installation, for example: /opt/aster), to leave vacuum for a file server.
- Mode of download of the configuration: no (for a file server), handbook (it is necessary to click on the button "To download now" to recover the configuration Aster of the waiter), automatic (astk questions the waiter with starting every 30 days).
- Last download: date of the last upgrade of information of configuration.
- Terminal: order to open a terminal on the waiter. This makes it possible to open a window of orders on the waiter when the bsf is used.
- Editor: editor text (for example, nedit). The procedure of installation chooses an editor among (and in this order): nedit, gedit, xemacs, emacs, xedit, VI.

The following values are turned over by `as_run --information` (nothing for a file server) and thus depend on the configuration of the `as_run` part:

- Type of platform.
- Versions available.
- Accessible machines: list of the accessible nodes of calculation since this waiter.
- Interactive batch/: specify if the waiter accepts launching in batch, interactive and provides the limits in memory, time CPU, many processors fixed on the waiter.

*The supported management software of batch is LSF, Sun Grid Engine and PROBLEMS.*

## 4.2 Configurations

### 4.2.1 General preferences

One reaches the window of configuration of the general preferences by the menu *Configuration general Preferences/* (see Figure 2.1.2-3).

This window makes it possible to inform the personal information of the user, to choose the language used by the interface.

Then, one finds the version which will be selected by default, the orders to reach a terminal and to an editor (as for the waiters).

**Many profiles in the small File** allows to preserve the name of the NR the last open profiles so as to point out them quickly.

**Level of message** indicate the level of details of the messages written in the window of the messages of the menu *Assistance*. Niveau=0: only messages <INFO> and <ERREUR> are written; the higher levels make it possible to debug the behavior of the interface. Level 1 is advised, it makes it possible to see the error messages being able to appear at the time of the problems of communication with the distant waiters.

For the follow-up of the jobs, one can choose the number of displayed lines when one visualizes the file output in the course of job (button **To search**), and the frequency of actualization of the list.

### 4.2.2 Preferences Network



One reaches the window of configuration of the preferences concerning the network by the menu *Configuration Preferences/Network* (see Figure 2.1.2-4).

This window makes it possible to inform the parameters network. One must then specify domain name network of the machine. For example, `domain.org` if the complete name of the machine is `mach00.domain.org`. If domain name is left empty, a message of alarm is displayed with starting because the names of machine indicated without domain name will not be supplemented. That can pose problems, but it is also sometimes necessary to leave domain name to vacuum in certain configurations.

If mode DHCP/VPN is active, it is in this field that one can indicate address IP of the local machine. While clicking on OK, the interface proposes IP addresses of the interfaces networks detected on the machine.

## 4.3 Tools

See Figure 2.1.3-1.

One selects the tool to be configured with the drop-down list, or a new tool is added. Standard tools are predefined (tiny the/capital ones are taken into account in the names of tools). *The tools are necessarily carried out on the machine "Room"* (where the interface is launched).

In general, the tools are launched either on a file Dyears mitre STUDY, is on a file when one traverses the filesystem with the bsf.

One defines simply the command line necessary to the launching of a tool (advised absolute way), one can place the codes @F, @R, @f, @D in the command line (see [§ 2.1.3]) to correctly pass a file to the tool. One can put these codes between brackets to be able to launch the tool alone, without file in argument.

Types of files can be associated with the tool. The type taken into account to launch the tool is the extension when one traverses the files with the bsf, the type of the drop-down list when it is about a mitre.

One can choose if the tool is usable on a distant file. In this case, astk is given the responsibility to bring back the file in question on the local machine in a temporary repertoire, of launching the tool, then to redeposit the file on the distant waiter (even if it were not modified by the tool).

## 5 How to make a study?

In this paragraph, one describes stage by stage how to use astk to carry out a study.

The study consists in calculating the answer of a thin cylinder under hydrostatic pressure. One has the following elements:

- The command file *Aster* : `forma00.COM`
- The description of the geometry carried out with Salomé : `formed00.datg`
- Grid tank built by Salomé : `formed00.mmed`

The following results are produced:

- A file result with the format med (fields of displacements,...): `forma00.resu.med`
- Classical files of message and Aster result: `formed00.mess` and `formed00.resu`

In the example, one places all the files in the repertoire `/home/tutorial`.

The files of this example are available in the repertoire `astest` of your version of *Code\_Aster*.

### Note:

*In the case of a study with several command files, all the files must be of type "COM", associated with the logical unit 1 and it is the order of appearance in the profile which determines the order of execution.*

### 5.1 Creation of the profile


One launches the interface which opens on a virgin profile, or if astk is already launched, one chooses *New file/* in the menu to create a new empty profile.

One places oneself in the mitre STUDY.


### 5.2 Selection of the files

#### 5.2.1 Definition of a basic way

In the mitre STUDY, one chooses a basic way to simplify the access to the files.

One clicks on the icon , the repertoire is chosen `/home/tutorial`.

#### 5.2.2 Addition of existing files

One adds the command file while clicking on , the selection of file opens directly in the basic way which one has just defined. It only remains to select the file `formed00.comm` (double-click or single click + ok), and the file appears in the list. Let us note that astk identifies the type of this file starting from its extension "COM", the logical number of unit is positioned to 1, the box "D" (given) is notched.

One makes in the same way for the file of grid to the format med (`formed00.mmed`). astk recognizes the extension "mmed", the logical number of unit is positioned with 20, the box "D" is notched.

One can also add the file `formed00.datg`. One strips the box "D", it will not be used in the study but one can visualize this grid by opening it with Salomé (see §5.5).


#### 5.2.3 Addition of files...

*Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.*

Copyright 2020 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

Except if an execution already took place, the files results do not exist yet, one cannot thus add them by traversing the tree structure.

### 5.2.3.1 ... while inserting a blank line


One clicks on , a line is added in the list. One chooses the type "mbe" in the list (what causes to position the logical number of unit with 6). The name is indicated /home/tutorial/formed00.mess or forma00.mess or . /formed00.mbe (since one can indicate the name of the file into relative compared to the basic way). The file is produced by the execution, one thus notches the box "R" (result) and one strips "D".

### 5.2.3.2 ... with "Value by default"

One could continue thus to add the other files, but one will use the function "Value by default" for the following files. This function uses the name of the profile astk to build the values by default (see [§ 2.2.1] /Menu contextuel), one will thus record the profile.

One chooses *To record under...* in the menu *File*, one goes with the browser in the repertoire /home/tutorial, and in the line *Selection*, one types formed00 (extension .astk is automatically added).

Let us note that the title of the principal window of astk gives the name of the current profile. The title is now: ASTK version 1.13.6 - formed00.astk - /home/tutorial

One inserts a blank line while clicking on , one chooses the type of file "rmed", then one clicks with the right button in the box of the file name and one chooses "Value by default": astk builds a file name starting from the basic way (see [§5.2.1]), name of profile (by withdrawing the extension) and of the type "rmed", that is to say /home/tutorial/forma00.resu.med. One sees as follows: . /forma00.resu.med (name relating to the basic way).

The box "R" was notched, and the logical number of unit fixed at 80. In the command file, one indicated:

```
IMPR_RESU ( FORMAT=' MED', UNITE= 80 ,...)
```

If one needs to add a new output file forma00.Rmed. ON proceeds in the same way : to select the type "libr", to modify the logical unit in 81 and the name of the file. The box "R" was notched, and the logical number of unit fixed at 80. In the command file, one indicated:

```
IMPR_RESU ( FORMAT=' MED', UNITE= 81 ,...)
```

ON thus modifies the logical number of unit consequently, it is enough to click on the old value, to erase it and type 81. Only two digits are displayed in this box, to avoid the errors, astk checks that the numbers of logical unit lie between 1 and 99.

In the same way, one adds a file of type "mess" and one of type "resu" in this manner (to leave the numbers of logical unit by default).

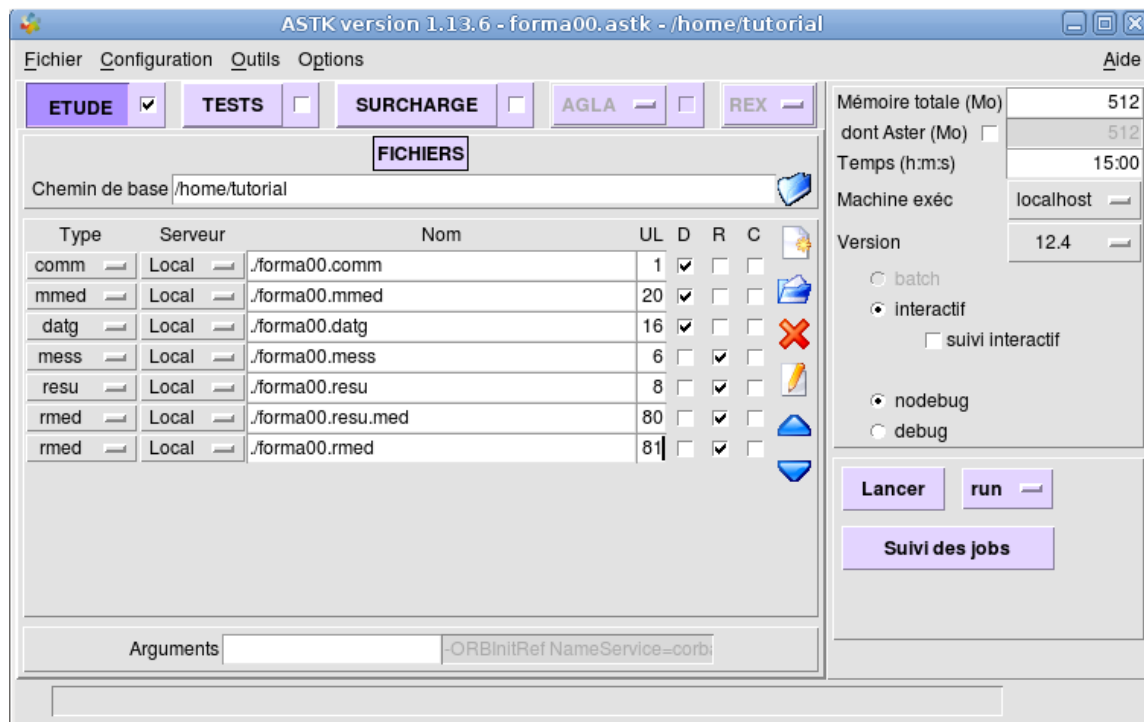



Figure 5.2.3.2-1: Window of the profile of study

## 5.2.4 To remove a file

To remove a line of the list of the files, it is enough to select it while clicking in the zone where the name of the file is indicated and to click on the icon .

### Note:

| Only the reference to this file in the profile astk is forgotten, the file itself is not unobtrusive!

## 5.3 Launching of calculation

The files given and results are selected, one adjusts the parameters of calculation (see [§ 2.3]), and one clicks on the button "Throw".

One takes care to notch the box which is just beside STUDY to announce that one wishes to use the contents of this mitre... if not answers us "Nothing interfaces it launching!".

If the profile were not recorded yet, the interface requires to choose a place and a name for this profile (see [§ 5.2.3.2]).

astk calls as\_run to carry out calculation, and transmits to the Follow-up of the jobs (asjob) the number of the job (number of the process in interactive) and other information which will make it possible to follow the advance of calculation. The initial state of calculation is HANGS (on standby), when calculation starts, it becomes RUN, then ENDED when it is finished (other states are possible in batch). The button "To bring up to date" calls the service which refreshes the state of calculations in progress.

When calculation is finished, one can consult the output of the job while double-clicking on the job, or by To publish/File output.

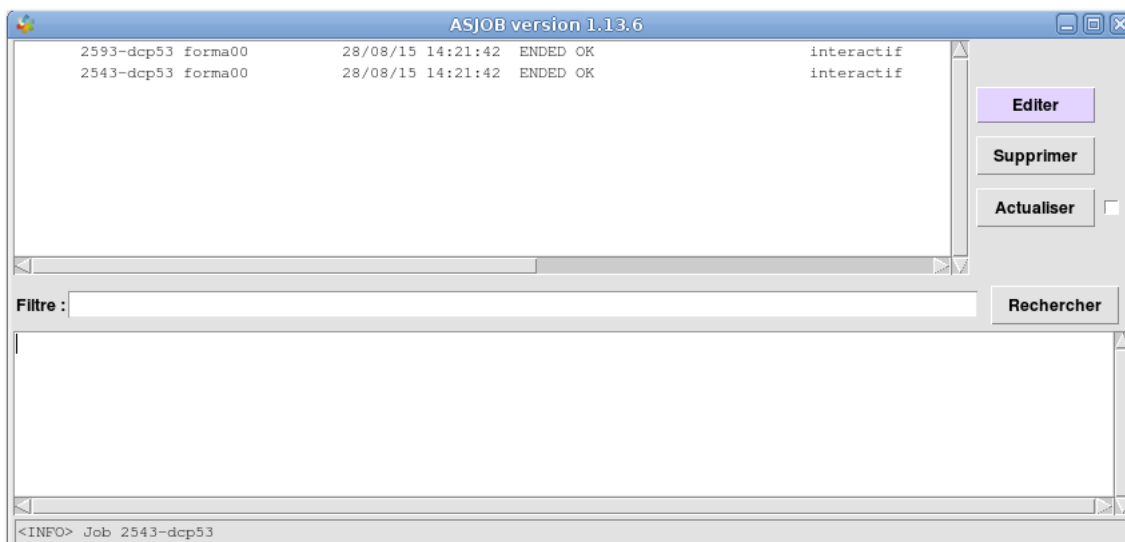


Figure 5.3-1: Window of follow-up of the jobs

## 5.4 Consultation of the results

One can consult the files results simply while double-clicking on their name, which opens a text editor for the files "mess" and "resu". If one have one result with the format grace, "dat", that causes to open this file directly in the grace graph plotter. L thus is visualizedbe curved (function displacement of time, stress-strain...) (provided grace was installed, and that "dat" is in the types of files associated with *grace*).

NB:

- The repertoire having to accommodate a file result does not exist, there is automatically created if the permissions are sufficient.
- If the copy of files results fails (problem of permissions, quota, distant connection...), they are copied in a temporary repertoire from the object computer. An alarm <A>\_COPY\_RESULTS show the way where it is necessary to go to seek the results.

## 5.5 Use of the tools

One can also use astk and the fact that one can freely define tools in it to gather in a profile all the files necessary to a study even if those are not directly used by *Code\_Aster*.

Example: To plot a curve with grace

The name is indicated `/home/tutorial/formed00.dat`. The file is produced by the execution, one thus notches the box "R" (result) and one strips "D" and the logical number of unit fixed at 21 . So in the command file one indicated:

```
IMPR_COURBE (FORMAT=' XMGRACE', UNITE=21, COURBE=_F (...))
```

One can open directly curve while making *To open with.../grace* (right click on the name of the file), to modify the geometry or the parameters of the grid.

Obviously, this is not limited to grace ; one can use other tools (maillieurs, tool of post - treatment,...) directly since astk and to reach thus all the files of a study since a profile with the adequate tool.

## 5.6 Advanced features

*Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.*

Copyright 2020 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

## 5.6.1 exectool

One chooses in `astk` the version of `Code_Aster`, the debug mode or `nodebug`, a possible overload and this will result in using such or such achievable `Code_Aster`.

One can be even further specifying the manner exact to launch this achievable: it is the role of the option of launching **exectool** (finely Options).

In ordinary time, `Code_Aster` is launched with a ordering of the type:

```
./aster.exe argument1 argument2...
```

To use the option `exectool` comes down to launching:

```
cmde_exec ./aster.exe argument1 argument2...
```

In small the Options, one can specify directly `cmde_exec` or a name of tool defined in the file of configuration of `as_run`: `[ASTER_ROOT] /etc/codeaster/asrun`

### Example 1:

In small the Options, one seizes in the box `exectool`: `time`

The ordering of launching will be thus `time aster.exe arguments...`

The order `time` accept exactly this kind of argument (achievable and its arguments), one will thus have the execution time of calculation. That does not have great interest, `Code_Aster` give already this kind of information.

### Example 2:

In the file of configuration `[ASTER_ROOT] /etc/codeaster/asrun`, one defines (on only one line):

```
memcheck: valgrind --tool=memcheck --error-limit=no --leak-check=full  
--suppressions=/opt/aster/valgrind-python.supp
```

It is then enough to indicate in small the Options, `exectool`: `memcheck`

`memcheck` is defined on the left in the file of configuration (of “: ”), therefore it is the complete order `valgrind...` who will be used during launching.

One can define tools as many as one wishes on condition that not entering in conflict with the other parameters preset this file. For this reason, the definition of the “exectools” would have advanced in the future.

## 6 How to launch a parametric study

One understands by parametric study a standard study (defined in the mitre STUDY) in which one wishes to vary one or more parameters.

The study is as general as any study defined in astk.

### Notice

*The study must be valid before being declined on the set of parameters. It must thus turn without error. It is also important to optimize the study before declining it on a large number of values of the parameters.*

The definition and the launching of a parametric study are described in [U2.08.07].

Launching is strictly identical to the launching of the nominal study only. Only the option "distribution" must be put at **yes** in small the Options.

Calculations are launched in parallel: to see §7.1 concerning the specific options.

## 7 Launching of calculations in parallel

### 7.1 Distribution of calculations

The management of distributed calculations is activated during the launching of a parametric study. Indeed, each calculation is independent of the different one. One can thus subject them in parallel to reduce the time of return.

Each unit calculation is subjected with the limits of times and memory indicated in the interface. By default, the main job will use the same limits. On the waiters of calculations, the parameters can be fixed (for example, 200 hours on Aster5) by using one *plugin* (see the documentation of development in [ASTER\_ROOT] /share/codeaster/asrun/doc/).

#### 7.1.1 Use of the available resources

One can insert in the profile (mitre STUDY), a file of the type "hostfile". One defines the list of the machines available in it and for each one the number of processors and the quantity of memory (out of Mo) usable.

Example:

```
[compute01]
cpu=4
mem=8192

[compute02]
cpu=1
mem=1024
```

That means that up to 4 calculations could be subjected on `compute01` (insofar as they do not ask more than 8192 Mo them 4) and 1 calculation on `compute02` using less than 1024 Mo.

In batch, one can subject much more calculations than of processors available and leave the manager batch distribute calculations on a cluster for example. In this case, one can fix `cpu=50` to leave to the maximum 50 calculations in the manager of batch.

If there is no file "hostfile" in the profile, one takes that whose name is fixed in the file of configuration [ASTER\_ROOT] /etc/codeaster/asrun under the label `interactif_distrib_hostfile` or `batch_distrib_hostfile` according to the mode of launching.

If no file "hostfile" is specified, the number of processors (hearts in fact) and the total memory are automatically given.

## Remarks

- One can easily collapse a machine while launching too many calculations with respect to the available resources. It is advised to get information about the use potential of means of shared calculations (class batch dedicated for example).
- Parallel calculations count well for the number of processors which they use and not for 1.
- Before the launching of calculations, connection to the nodes of calculation is tested. The list will be limited to the machines having been contacted successfully.

## 7.1.2 Time of expiry

When the number of calculations to launching is much higher than the number of processors overall available (and it is often the case), of calculations are on standby of tender.

If a calculation requires more memory than no machine can offer some, it would remain indefinitely on standby.

To avoid that, a time of expiry (time-out) is definite equal at the time of main calculation, i.e. the time chosen in astk during the total tender.

If no calculation were subjected during this time, calculation is rejected.

## 7.2 Activation of the parallelism of Code\_Aster

Parallelism interns with *Code\_Aster* is present in two forms:

- OpenMP parallelism functions in shared memory and is available inS solvorS `MULT_FRONT` and `MUMPS`. It is necessary of course that the version of *Code\_Aster* was compiled with the adequate options.
- Parallelism MPI (by sending of message, Passing Message Interfaces) is available inS solvorS `MUMPS` and `PETSC` and in elementary calculations. Compilation is much more complicated and is not automatic at the time of the installation of *Code\_Aster* (it is necessary to choose an implementation MPI, to compile them pre-necessary and in particular `MUMPS` in MPI, then *Code\_Aster*).

One chooses the number of processors used in OpenMP and the number of processors used in MPI (distributed on a certain number of nodes of calculations) in small the Options (see §2.1.4).

## 7.3 Multiple executions

It is about a mode of quite particular launching bound for the developers.

The objective is simultaneously to carry out a profile (a study or a list of CAS-tests) on several machines.

One activates it while notching `multiple=oui` in the menu `Options`. Then, at the object time a window opens in order to select (while notching) the machines on which the study or the tests will be started.

Results, including the output files *output* and *error* usually copied in the repertoire `flashor`, are copied in the repertoire `$HOME/MULTI` (`$HOME` being in general equal to `/home/username`). One can choose to leave the results on each machine, which is advised if the files are overall bulky, or to repatriate all the files on the local machine.

Obviously, there are some precautions to take so that functions: the selected version must be available on all the machines, the parameters of calculation compatible with the resources of each machine, etc.



## 8 Use of as\_run

When one uses the interface askt for launching calculations, this one as a customer calls on proposed services by a waiter which can be on the same machine or a distant machine (if the waiter is distant, there are file-swappings and a Shell order through the network (protocol rsh or HS) that we do not detail here).

Features of `as_run` can be classified in several categories:

- **for the user** : features which can be called by **users**, such as for example launching *Code\_Aster* "with the hand",
- **for the developer** : features used by the developers to visualize a file, to copy a file, to build a list of tests, to check the catalogue of messages...
- **reserved to the customers** : irrelevant features direct out of askt (or another customer),
- **for the tasks of administration** : link towards the tool of follow-up of the anomalies, the database of studies...

Various features of `as_run` are given while typing: `as_run --help`.

In the files of configuration of `as_run` and in the files `.export`, one can use Dbe variable D'environnement (only for the file names).

For example, in `etc/codeaster/asrun` :

```
mpirun_cmd: $ASTER_ROOT/public/mpi/bin/mpirun...
```

or in a file `.export` :

```
R repe $HOME/results_repe R 0
```

### 8.1 For the user

If the order `as_run` is not available directly, it is necessary to carry out:

```
Salome Shell -- as_run [options] [arguments]
```

Salome being a script available in the installation of Salomé-Meca. One can find it while replacing `nnn` by the index of version in: `/opt/salome-meca/nnn/appli_Vnnn/salome` .

- Launch the execution described by the profile (action by default):

```
as_run --run [options] nom_profil.export
```

The file export can be in room or on a distant machine.

Each line of the file starts with:

- P: definition of a parameter,
- A: definition of an argument of the line of orders of *Code\_Aster*,
- F: definition of a file,
- R: definition of a repertoire,
- NR: used only for the workshop of software engineering (AGLA).

The format for the parameters and arguments is: `P nom_parametre value or With nom_argument value`.

For the files and repertoires, the format is:

```
Standard F/R unite_logic way DRC
```

where `DRC` specify if the file or repertoire is in `Donnée`, `Résultat` (both are possible unit), and if the contents are `Compressé`.

- Quickly carry out a calculation starting from the files in arguments:

```
as_run --quick [options] file1 [file2 [...]]
```

The developers can indicate an overload of sources FORTRAN (resp. python) with the options `--surch_fort` (resp. `--surch_pyt`).

- Carry out a CAS-test in interactive:  
`as_run --test [options] testcase [results_directory]`
- Product a named dynamic library FILE by compiling the source files src1... srcN. Used typically to build a library UMAT. :  
`as_run --make_shared --output=FILE [src1 [...]] srcN`

It is the compiler and the options of compilations defined in the file `config.txt`, version by default or selected with the option `--towards`, which is used to build the library. FILE is the name of the produced library which will be indicated in the keywords of *Code\_Aster*.

## 8.2 For the developer

### Notice

*[These functions are not maintained any more and could disappear.*

- Poster a source file: FORTRAN, C, python, capy, catastrophes, histor or test:  
`as_run --show [options] obj1 [obj2...]`
- Copy a source file in the current directory:  
`as_run --get [options] obj1 [obj2...]`
- Poster the diff of a source file: FORTRAN, C, python, capy, catastrophes, histor or test:  
`as_run --diff [options] obj1 [obj2...]`
- Poster the principal subroutine of an order *Code\_Aster* :  
`as_run --showop [options] order [.capy]`
- Built a file export to launch a CAS-test and prints it on stdout:  
`as_run --get_export testcase_name`
- Built a list of CAS-tests starting from orders/keywords and/or checking criteria of time CPU or memory:  
`as_run --list [--all] [--test_liste=FILE] [--filter=...]  
[--command=...] [--user_filter=...] [test1 [test2...]]`
- Built the diagnosis of the CAS-tests *Code\_Aster* (since DIRi or the repertoire `astest` by default) and a file pickled result writes:  
`as_run --diag [--astest_dir=DIR1, [DIR2]] [--test_liste=LISTE]  
[--only_nook] [diag_result.pick]`
- Operation on the catalogues of messages of *Code\_Aster*. `subroutine` = turns over the messages called by "subroutine". `message_number` = turns over the routines using this message. `check` = checks the catalogues and displays some statistics. `move` = moves a message of a catalogue with another and produces the modified catalogues and source files. :  
`as_run --messages subroutine | message_number  
as_run --messages check [--fort=...] [--python=...] [--unigest=...]  
as_run --messages move old_msgid new_msgid`
- Turn over information on the processors and the memory of the machines given:  
`as_run --get_informations [--output=FILE] host1 [host2 [...]]`

## 8.3 To maintain an installation local

- Turn over the number of the version of development:  
`as_run --getversion [options]`

- Turn over the way of installation of the version of development:  
`as_run --getversion_path [options]`
- Poster of information in connection with the installation and the configuration:  
`as_run --showme [options] bin|lib|etc|dated|local|to rmdir`  
`as_run --showme [options] param parameter_name`

## 8.4 For the interfaces launching of calculations

- Launch an execution (by calling `as_run` in a separate process):  
`as_run --serv user@mach: /nom_profil.export`
- Send the contents of "filename" (possibly on a distant machine) to the addresses email1, email2,...:  
`as_run --sendmail [--report_to=EMAIL1, EMAIL2] filename`
- Turn over information on the configuration of a waiter: batch, interactive (yes/not, limits), nodes of calculation, versions:  
`as_run --information`
- Turn over the state, the diagnosis, the node of calculation, time CPU and the repertoire of work of a calculation:  
`as_run --news job_number job_name mode`
- Open the file output or error on the provided screen:  
`as_run --edict job_number job_name output mode|error DISPLAY`
- Poster end of the file fort.6 or lines corresponding to the filter:  
`as_run --tail job_number job_name mode fdest nb_lines [regexp]`
- Stop a calculation and destroys the associated files:  
`as_run --del job_number job_name mode [node] [--signal=...]`
- Remove the files of calculations which are not in the list:  
`as_run --purge_flash job_number1 [job_number2 [...]]`
- A file (possibly distant) with format HTML converts and writes the result in FILE:  
*in beta release*  
`as_run --convert_to_html [user@machine:] file --output=FILE`

## 8.5 For the tasks of administration

- Insert a new entry in the system of follow-up of anomalies and copy the attached files if a file export is provided:  
`as_run --create_issue issue_file [export_file]`
- Fills the fields "corrudev" or "correxpl" (according to worms) in the cards found in 'histor 'and possibly the farm:  
`as_run --close_issue --vers=VERS histor`
- Extract contents of the cards listed in 'input\_file 'in 'histor ':  
`as_run --extract_histor [--status=STAT] [--format=FORMES]`  
 `[--all_msg] input_file histor`
- Prepare the profile to insert an execution in the database:  
`as_run --insert_in_db [export_file]`

## 8.6 To supervise the launching and the return of a calculation

One provides here methodology for launching calculations since a local machine towards a waiter of calculation, except ihm graphic, by calling the various services of `as_run`.

The explanations provided hereafter allow:

- to launch a calculation,
- to supervise a calculation (to know its state),
- to recover the results,
- to publish the files output/error.

### 8.6.1 Assumptions

One explains the various stages by taking the example of the execution of a calculation on the Aster5 waiter since a Calibre7 machine. The data files must necessarily be on the station Gauges, that which subjects calculation, or on the waiter of execution (`.export`, data files, results, base,...).

In the examples, the following names of machines are used (and thus to adapt to each situation):

- Calibre7 (customer): `hostname=dspXXXX`
- Aster5 (waiter): `hostname=aster5.hpc.edf.fr`, `user=NNI`

The heading used in the files `.export` to identify the waiter (it is necessary to modify `Ni` and `dspXXXX`):

```
P mclient dspXXXX
P waiter aster5.hpc.edf.fr
P username Ni
P protocol_exec asrun.plugins.server.SSHServer
P protocol_copyto asrun.plugins.server.SCPServer
P protocol_copyfrom asrun.plugins.server.SCPServer
P aster_root /home/rd-ap-simumecca
P to_proxy_dir /scratch/NNI
P platform LINUX64
```

Preliminary checking: Connection on command line `HS aster5.hpc.edf.fr pwd` must function without password.

### 8.6.2 Orders

The addition of the option `--nodebug_stderr` remove the impression of the exit `stderr` of `as_run`.

#### 8.6.2.1 Recovery of information of the waiter (permet also of to test connection)

```
as_run --nodebug_stderr --proxy --information info.export
```

with `info.export` who does not contain that the heading.

In particular, one can see the list of the versions available. Then, the version will be used `testing`.

#### 8.6.2.2 Launching of a calculation

```
as_run --nodebug_stderr --proxy --serv study.export
```

with `study.export`:

```
<<< makes dizzy >>> +

P actions make_etude
P nomjob jobname
```

```
P origin ASTK 1.13.4
P batch mode
P consbtc yes
P soumbtc yes
P version testing          <<< provided by --information
P tpsjob 10
To memjeveux 512
With tpmx 600
P memjob 4194304
F COM $HOME/dev/codeaster/src/astest/zzzz100f.comm D 1
F mess /tmp/zzzz100f.mess R 6
F resu zzzz100f.resu R 8
R bases NNI@aster5.hpc.edf.fr: /home/Ni/study1/base R 0
```

/tmp/zzzz100f.mess will be recopied on the local machine.

zzzz100f.resu will be recopied on the local machine in the repertoire where was carried out --serv.

It is necessary to add NNI@aster5.hpc.edf.fr: in front of the name to specify that these files are already on the waiter (or will have to remain there as a result, case of the base in the example).

### 8.6.2.3 Interrogation

Then, one can use the same file .export for the various actions, the last parameters will be ignored most of the time, post.export :

```
<<< makes dizzy >>> +

P jobid 12399454          <<< provided in the output of --serv
P batch mode
P nomjob jobname

P tail_nblne 50          <<< only for --tail

P edit_type output       <<< only for --edict
```

### 8.6.2.4 Actualization of the state of the job

```
as_run --nodebug_stderr --proxy --news \  
--schema=asrun.plugins.default.actualité_simple post.export
```

The diagram by default makes \*actuality + get\_results\*. If one wants to make only actualization, it is necessary to indicate --schema=asrun.plugins.default.actualité\_simple. If one wants to make the actualization and the recovery of the results, one makes simply:

```
as_run --nodebug_stderr --proxy --news post.export
```

### 8.6.2.5 Recovery of the results

```
as_run --nodebug_stderr --proxy --get_results post.export
```

### 8.6.2.6 To remove the job

Calculation stops if it is in hand, removes the files of the flasher :

```
as_run --nodebug_stderr --proxy --del post.export
```

## 8.6.2.7 To consult the end of the output of a job in progress

```
as_run --nodebug_stderr --proxy --tail post.export
```

## 8.6.2.8 To publish the file output

```
as_run --nodebug_stderr --proxy --edict post.export
```

Possible values for edit\_type : output, error, export, script, diag.

## 8.7 Options

```
--version          show program' S version number and exit
- H, --help        show this help message and exit
- v, --verbose     the program makes more talkative
--silent          function as silently as possible
- G, --debug       poster information of debug
--stdout=FILE     allows to redirect the usually written messages
                  on sys.stdout
--stderr=FILE     allows to redirect the usually written messages
                  on sys.stderr (only messages of asrun)
--log_progress=FILE redirect information of advance towards a file
                  instead of sys.stderr
--nodebug_stderr  remove the impression of information of debugging on
                  stderr
- F, --force       force the operations which can use a mask
                  (download, compilation...)
--display=DISPLAY value of the variable DISPLAY (NB: some
                  functions read this value in a file)
--rcdir=DIR       use the repertoire of resources $HOME/'DIR' (it
                  defect is .astkrc). It is necessary to avoid indicating
one
                  absolute way because it will have passed to the distant
waiters.
--remote_shell_protocol=REMOTE_SHELL_PROTOCOL
                  protocol used for the launching of orders
--remote_copy_protocol=REMOTE_COPY_PROTOCOL
                  protocol used for the copy of files
--copy_all_results Copy all results in the current directory (for --quick
                  action)
--proxy           request with a waiter to launch the action specified (by
                  example, by calling as_run --serv on the waiter)
--schema=SCHEMA  allows to modify the behavior of as_run in
                  using an alternative diagram
```

Options for the maintenance actions:

```
--filter=FILTER   filters applied to the parameters of the tests: 'will
nom_para         < valeur' (supported comparisons <, >, =).
--vers=VERS       Version of Code_Aster to be used (for get, show,
                  showop)
--force_upgrade  Force the update towards the next version (by
                  example of 10.1.xx towards 10.2.0)
- O SLIPS BY, --output=FILE
                  redirect the result in FILE instead of stdout
--surch_pyth=REP one or more repertoires (separated by a comma)
                  containing the added files python
--configuration=FILE use another file "config.txt" (only for
                  make, update and auto_update).
```

*Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.*

Copyright 2020 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

```
--only_nook          bring back only the errors (the time spent in
tests ok is counted)
--surch_fort=REP     one or more repertoires (separated by a comma)
containing added files FORTRAN
- has, --all         copy of all the files of the CAS-test
--destdir=DIR       root directory (fake-root) where the files will be
copied
--keep_increment     update a version increment by increment in
preserving the achievable files intermediary
--search=REGEXP     preserve the tests checking the regular expression
data (or a simple character string).
--unigest=FILE       a file unigest (for the suppression)
--command=COMMANDE  [/MOTCLEFACT [/MOTCLE [=VALEUR]]]
preserve the tests using the order and words
keys given
--test_liste=FILE   list of the CAS-tests
--report_to=EMAIL   address of transport where to send the report of one
execution (only for --auto_update)
--user_filter=FILE  file containing the classes testlist.FILTRE. See
[...] /share/codeaster/asrun/examples/user_filter.py
like example.
--astest_dir=DIR    repertoire of the CAS-tests to be analyzed
- L, --room         the files are not sought on the waiter
distant but on the local machine
--nolocal           force the research of the files on the distant waiter
(opposite of --room)
```

Options for operations one jobs:

```
--signal=SIGNAL     send a signal to calculation (KILL|USR1)
--result_to_output  Redirect the result in FILE instead of stdout
```

Options for the interface with the REX:

```
--status=STAT       an error occurs if the cards are not in
this state
--all_msg           recover all the messages of the cards
--format=FORMES     format of the generated histor (text or HTML)
```

## 8.7.1 Additional precise details

- When one recovers a source file with `get`, `show` or `showop`, this one is put in `/tmp/astk_login /cache`. If for the same file again is asked, this one is taken directly in this repertoire except if the option `--force` is activated; the files which have more than one day are automatically removed mask.
- `--room/--nolocal`: these options are used when the behavior by default is not appropriate. In the use of `show/get`, it is preferable to reach the sources buildings, it is thus the defect. In this case, one will use `--nolocal` to reach a version on the waiter of reference. During the maintenance of a local installation, the defect is to download the updates on a distant waiter. Nevertheless, if one to recover the packages locally, one can use `--room` to install them.
- The principal file of configuration is `[ASTER_ROOT] /etc/codeaster/asrun`, the user can define certain values in `$HOME/.astkrc/prefs` (where `$HOME` is the repertoire by default of the user), in particular user name to be used on the distant waiter (`devel_server_user`) to avoid an alarm with each launching of `as_run`.
- Complements on the inner working of `as_run` (client/server mode, "diagrams/plugins") or on certain options are available in the documentation developer embarked with `as_run` in `[ASTER_ROOT] /share/codeaster/asrun/doc/`.

## 9 Frequently asked questions

---

***“Nothing occurs when one tries to launch a calculation, to publish a file or to open a terminal on a distant waiter” or “Code return = 2, Profiles Copy failed in the window of the messages to the launching of a calculation”***

It is probable that the local machine cannot communicate correctly with the distant waiter. Communications using protocol CCP/rsh, files `.rhosts` on the local machine and on the distant waiters must be correctly well informed. If one uses protocol SCP/HS, it is necessary that the private and public keys are coherent.

One can check it by leaving `astk`, and by starting again it in this manner:

```
astk --debug 0 - - check      (--debug 0 is used to limit the impressions)
```

To attentively see information which is printed. `astk` provides information on the variable of environment `DISPLAY` (to check that it is correctly defined), and tests connection between the machine where `astk` is launched and the waiters distant. In the event of problem of connection with a waiter, `astk` suggests a modification of the file `.rhosts` target machine.

***“When a calculation in batch is subjected, there is the message: The number of the job and the class could not be recovered”***

Time or the memory required is probably beyond the limits of the classes of work. See in the window of the messages, there is probably a message of the manager of batch of the style “Boat exceed queue' S hardware limit (S)”.

To decrease time or the memory, or to explicitly choose a class batch which agrees in the additional options.

***“While clicking on Value by default the file name remains empty”***

This function uses the name of the profile to determine a value by default, it is necessary to record the profile before being able to use this function.

### ***To submit a report of bug:***

If an error remains unexplained, here method to submit a report of bug while providing Lbe traces of execution.

1. To close `astk` if it is in execution and to start a new session (that makes it possible to limit the traces at least relevant).
2. To reproduce the problem by making the minimum of thing.
3. To leave `astk` and join to the support the file `$HOME/.astkrc/log.astk` (or `.astkrc_salomeXXXX` in Salomé-Meca).