Interface access to Code_Aster : astk

Abstract:

This document presents astk (contraction of Aster and Tk, decides “astek”), the interface making it possible to organize its computations Code_Aster.
One describes the logic of operation of astk, the graphic interface itself, bsf which makes it possible to explore the files on the various servers, then one explains how the user can configure astk according to his needs.

For a rapid access with Code_Aster, one shows from examples how to launch a study and how to make a parametric study.

For the people having access to the server of computation Code_Aster of the EDF network, one explains how to emit one anomaly file, to reach the files back from experiment (REX).

Lastly, in one second part, one presents as_run which gives many opportunities on command line.

The features of astk version 1.13 are described here which manages all the supported versions of Code_Aster.
Contents

1 Presentation and notions générales 4
   1.1 Démarrage 4
   1.2 Fonctionnalités 4
   1.3 Mode of fonctionnement 4
   1.4 Servers astk 5
   1.5 Profile astk 6

2 Description of the interface graphique 7
   2.1 Bar of menu 7
      2.1.1 Small Fichier 8
      2.1.2 Small Configuration 9
      2.1.3 Small Outils 11
      2.1.4 Small Options 12
      2.1.5 Small Aide 13
      the 2.2 onglets 14
         2.2.1 Mitre ETUDE 14
            2.2.1.1 List types for ETUDE 15
            2.2.2 Button REX 16
      2.3 Parameters of soumission 17
      2.4 Bar of état 19
      2.5 Followed by the jobs 19

3 Limps of selection of files: bsf 20
   3.1 Navigation 21
   3.2 Small File and small contextuel 22
   3.3 Small Affichage 22

4 Configuration 23
   4.1 Serveurs 24
   4.2 Configurations 24
      4.2.1 Générales 24 Preferences
      4.2.2 Preferences Réseau 24
   4.3 Outils 25

5 How to make a study?  26
   5.1 Creation of the profil 26
   5.2 Selection of the fichiers 26
      5.2.1 Definition of a path of base 26
      5.2.2 Addition of files existants 26
      5.2.3 Addition of files 27
         5.2.3.1... while inserting one line vide 27
         5.2.3.2... with “Value by defaults” 27
      5.2.4 To remove a fichier 28
5.3 Launching of the calcul

5.4 Consultation of the résultats

5.5 Use of the outils

5.6 Features avancées

5.6.1 exectool

6 How to launch a study paramétrique

7 Launching of computations in parallèle

7.1 Distribution of calculs

7.1.1 Use of the resources disponibles

7.1.2 Time of expiration

7.2 Activation of parallelism of Code_Aster

7.3 Executions multiples

8 Use of as_run

8.1 For the utilisateur

8.2 For the développeur

8.3 to maintain an installation locale

8.4 For the interfaces launching calculs

8.5 For the tasks of administration

8.6 To supervise the launching and the return of a calcul

8.6.1 Hypothèses

8.6.2 Commandes

8.6.2.1 Recovery of information of the server (also allows to test connection)

8.6.2.2 Launching of a calcul

8.6.2.3 Interrogation

8.6.2.4 Actualization of the state of the job

8.6.2.5 Recovery of the résultats

8.6.2.6 To remove the job

8.6.2.7 To consult the end of the output of a job in cours

8.6.2.8 To publish the file output

8.7 Options

8.7.1 Precise details supplémentaires

9 Questions Fréquentes
1 Presentation and general notions

the provision of tools as Code_Aster which functions on multiple platforms requires a graphic interface to simplify the work of the user.

In addition, the evolution of the centralized layers of computation and the 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 Aster computations: to prepare the data, to organize the files, to reach the tools the pre one and postprocessing, to launch and follow the evolution of computations. astk also makes it possible to choose the version of Code_Aster to be used among those available (stable, testing…).

The name “astk”, to pronounce “astek”, comes from the contraction from Aster and Tk which is the graphic library used.

1.1 Starting

In Salome-Meca, to open the modulus Aster, then to launch astk since its icon or since the small Aster/Tools/Run ASTK.

Apart from Salome-Meca, the command astk, in general, was placed in the path by default of the users, the interface opens with a configuration by default (see [§ 4] Configuration).

If the command is not found, it is necessary to launch: [aster_root] /bin/astk where [aster_root] is the directory of installation of Code_Aster (contact possibly your administrator).

1.2 Features

STUDY: astk makes it possible to launch an Aster computation on the local machine (for example in the frame of a use on personal computer), on a server of departmental computation or of course the server of Aster computation of 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. Files back from experiment can be emitted and consulted by the users since the interface, the developers can answer it (access to necessary the EDF server). This functionality is reserved at EDF and to its people receiving benefits.

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

The transition of a firebreak is not proposed.

TOOLS: The user can launch various predefined tools and configure those which it needs (mesh generator, 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, printing, or to open a window of commands from these machines.

ASJOB: One can follow computations, 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 and the authorizes a clear separation between the interface (customer) tools used to reach the code (services). The entrance point of the server 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.
The communication protocol by default between the various machines is HS for commands Shell 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 command `astk` or `codeaster-GUI`) asks for the launching of a computation on a server of computation.
The interface carries out a service `as_run` which is given the responsibility to copy the data files from the server (in an intermediate directory defined by this one and divided by all the nodes of this server), to require of the server to start computation. The results of computation remain on the server of execution (in the intermediate directory if they are supposed being recopied towards the customer or in their final directory if their destination is on the server of computation). The results are repatriated if necessary on the machine customer when state of computation is ENDED in the follow-up of the jobs; i.e. after a manual or automatic actualization.

So only connection in the meaning machine customer towards server is requested (key valid HS for a connection without password). There are no more the connections opposite initiated by the server towards the customer (as in the previous versions to the 1.9).
Moreover, once subjected computation, connection network between the customer and the server can be stopped without risk to lose computation because the data were already deposited on the server and the results will be repatriated only on request of the customer (interface).

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

### 1.4 Servers astk

For astk, a server is:

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

The server 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 computations on the local machine (on which are installed a version of Aster and the part server of astk), it is also necessary to declare this machine like a server of computation (Local and Machine in the following example and §2.1.2)

In this document, one will call “server of reference” the machine on which the qualified version of in-house Code_Aster EDF is managed. Certain features are accessible only if the user has an access to this machine.

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

astk is used on the machine of name `mach00`, on which Code_Aster is installed. One has access to a departmental file server `file01`, the files of this server are accessible (assembly NFS) since two machines from computation `comp02`, `comp03`. One N has also access to a cluster with nodes of computation whose frontal machine is `front04`.

One has then:

- “Local” (reserved label): one has it always to explore the files which are on `mach00`;
- “Machine” (unspecified label, the procedure of installation fixes this name at the value turned over by the command “`uname - N`”): whose L” addresses IP is that of `mach00` which is the server of computation (services `as_run` installed) of the local machine;
- “Departure” (unspecified label): of which the IP address is that of `file01`, which makes it possible to explore the files lodged by the file server, configured like a server of Aster.
computation (services as_run installed) having comp02 and comp03 seen as of the nodes of computation (comp0i can be the same machine as file01);

- “Cluster” (unspecified label): of which the IP address is that of front04 which is a server of computation (services as_run installed) having N nodes, the only accessible machine being front04.

The difference between “Departure” and “Cluster” is that in general the stations of computation are accessible directly (one could thus subject a computation 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 servers of computation “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, as a result, the type associated with each file, parameters of submission of computation (memory, time, machine of computation, 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 Record under…) and with each launching of a computation (it is the file whose extension is, by convention, .astk).

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

The graphic interface (GUI) breaks up into 4 parts:

1. A menu bar
2. The mitres
3. The parameters of submission
4. The bar of state
5. Zones arguments Aster

Figure 2-1: Principal window

The mitres TESTS, SURCHARGE 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

- **File**: Allows to create a new profile.
- **To open**: Open a profile created before by astk.
- **To record**: The profile saves running (this is done automatically during the launching of a computation).
- **To record under**: The profile saves running while giving the choice of the name of the profile and its site.
- **To copy under**: Copy the profile running in another directory and proposes to copy the files (in data and/or results) that it reference.
- **To import**:
  - `.export astk`: Reads again a file `.export` of a preceding execution (that can be the file *.pNNN directory $HOME/flashor).
  - `.export astk (addition mode)`: contrary to the classical importation which starts from a virgin profile, the files and directories 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 file REX`: Import the files attached to a file of the REX which one gives the number.
  - `benchmark`: Import the files necessary to the launching of a benchmark. The files are taken on the selected object computer.
  - `benchmark (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` of the current profile.
- **To close**: To close the profile running
- **To leave**: Finish the application
- **N** the last open profiles are directly accessible starting from the small File.
2.1.2 Small Configuration

- **Servers**: Allows to modify the configuration of the servers: servers of computation or file servers (see Figure 2.1.2-2).

  For the servers of computation, one chooses the mode of download of the configuration Aster: the server turns over the list of the versions available, the procedure (batch and/or interactive) and the limiting associated.

  It is necessary to indicate the "login" to use to connect itself to the server, and the directory where *Code_Aster is installed*.

  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 restrain towards the management tools of feedback (emission, consultation of the files 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 personal choice, sorted or not by name (see §2.2.1 2.2.1).

    For the follow-up of the jobs:
    
    The number of lines displayed during the consultation of a computation in the course of execution (tail), and the automatic frequency of actualization in minutes.

  - **Networks** (See Figure 2.1.2-4)
    
    the domain name network of the machine and if mode DHCP/VPN (addresses dynamic IP) is active, possibilities of fixing variable DISPLAY for the display of the external applications, the communications protocols used for the commands Shell (rsh or HS) and copies it 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 option DISPLAY with full knowledge of the facts and only if the default value 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 the domain name…).

- **Colors**: Classic, KDE3 Hook, Nostalgic, Personalized,…

- **Police**

- **Records** the position and cuts windows
Figure 2.1.2-2: Small Configuration/Servers

Figure 2.1.2-3: Small Configuration/General Preferences/
Remarks

That one chooses protocol rsh/CCP or HS/SCP, connections must function without password: files .rhosts correctly filled or private/public words valid SSH.
According to what turns over the command “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 command… In this case, the commonplace connection of the machine towards itself must also be configured.

2.1.3 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 for postprocessing, orders STANLEY,
  - Converter of mesh: Allows to convert a mesh of a format towards another, the formats available are: aster, med, gmsh, ideas and gipi,
  - To convert with HDF format: Allows to convert a classical base with HDF format Aster,
  - bsf: Explorer of files multi-machines,
  - Eficas: Editor of file of Aster commands,
  - Gipi: Mesh generator and tools of postprocessing (usable free with Aster),
  - Gmsh: Mesh generator and tools of postprocessing (free),
  - grace: Graph plotter 2D.

When throw Stanley is chosen, astk seeks in the profile the bases available (by order preferably a base as a result, if there is not, a bhdf (HDF bases) as a result, if not a data base and finally, a bhdf in data), produces a temporary profile starting from the profile running with a file of Aster commands which starts with POURSUITE () 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 tools and to specify if one can use the tools on a distant file.

The following codes can be used in the command line:
  - @F: absolute path of the selected file,
  - @R: directory containing the selected file,
  - @f: name of the file (without the directory),
  - @D: addresses 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 tools to be used lean 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

![Small Options Arguments](image)

- : 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 crunchings and destruction of objects,
  - `rep_outils`: the directory defines where the tools called since Aster (onmetis or gibi for example) are sought,
  - `rep_dex`: the directory defines where are sought the external data (given mesh),
  - `rep_mat`: the directory defines where the data of the catalog 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 with 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. The three first are used during launching of tests.

---

**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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
Code_Aster

Version default

Titre : Interface d’accès à Code_Aster : astk
Date : 07/07/2015 Page : 13/41
Responsable : Mathieu COURTOIS
Clé : U1.04.00 Révision : 13195

nbmaxnook : maximum number of invalid benchmarks (NOOK, ARRET_ANORMAL...) before the interruption of the list of the tests,

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

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

RESNOOK : one keeps only the files of the 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 executable built in debug mode, slower.

corefilesize : cut limit of the file core during a planting

Options of launching

ncpus : the number of processors used by solver MULT_FRONT mpi_nbcpu

defines : the number of processors for parallelism MPI mpi_nbnoeud

defines : the number of nodes for parallelism MPI defines (Where the mpi_nbcpu processors will be distributed)

classifies : allows to choose the class batch (or groups it of class) in which computation 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 computation. The time of departure is provided to the format [[month:] day:] hour: distribution

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

exectool : launching by means of particular tools (see §5.6.1)

multiple : carry out the profile on several machines (value yes/not, to see §77).

2.1.5 Small Assistance

• Assistance/Small Parameters of computation : access to the text of Parameters
• assistance of computations : allows to define the size memory, the version of Code_Aster which will be used, the way of calculating (Batch or interactif),...
• Small File/Configuration/tools/options...
• History of the modifications : Evolution of astk 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), one obtains information more or less.
• By the way… : the impossible to circumvent window of information.

2.2 The 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 library (and one did not want to add pre-necessary additional!).

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
STUDY is a mitre in which one informs the list of the files necessary to launch a study. REX is a simple button which 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 directories), or to simplify the display of the file names, one can define a basic path.

![Figure 2.2.1-1: Basic path](image)

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

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Server</th>
<th>Nom</th>
<th>UL</th>
<th>D</th>
<th>R</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>comm</td>
<td>Local</td>
<td>/demo001a.comm</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>datg</td>
<td>Local</td>
<td>/demo001a.datg</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Figure 2.2.1-2: List files and directories](image)

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

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

When one changes the type of a file/directory, default values 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” thus makes it possible 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 directory for such type.

To handle the list, one has six buttons:

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

 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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
• move line downwards.

Arguments:
This zone of text makes it possible to transmit arguments to the executable Aster. See also \[\S2.1.4\].

Finely contextual:
While clicking with the right button on an entry of the list, one reaches a contextual menu:
• To open: launch the tools associated with this kind of file (determined by the “Standard” field, and not the extension of the file), if no tools are associated with this type, one publishes the file;
• To publish: publish the file (or all the files of the directory) in the same way as the button ;
• To open with…: one can choose to open the file selected with one of the tools available (result can be astonishing if the tools do 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 not proposed if the profile were not recorded yet (is used in general on one line virgin which one has just inserted, indicators D/R/C are not affected by this operation);
• Terminal: a final window xterm Properties makes it possible • to open: display the permissions, the size, the date and the hour of the file (command ls it).

2.2.1.1 List types for STUDY

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM</td>
<td>files of Aster command (including the files of poursuite)</td>
</tr>
<tr>
<td>mail</td>
<td>file mesh with the Aster format</td>
</tr>
<tr>
<td>wanders</td>
<td>file of error (fort.9 of Aster)</td>
</tr>
<tr>
<td>mess:</td>
<td>file of the messages of the execution</td>
</tr>
<tr>
<td>resu:</td>
<td>results file (printing of the tests, printing to the Aster format )</td>
</tr>
<tr>
<td>bases:</td>
<td>directory containing the base of computation</td>
</tr>
<tr>
<td>bhdf</td>
<td>directory containing the base of computation with HDF format</td>
</tr>
<tr>
<td>cast:</td>
<td>results file with format CASTEM</td>
</tr>
<tr>
<td>mast</td>
<td>kept for reason of compatibility</td>
</tr>
<tr>
<td>mgib:</td>
<td>mesh with the format Gmsh</td>
</tr>
<tr>
<td>mmed:</td>
<td>mesh with med format</td>
</tr>
<tr>
<td>the msh:</td>
<td>mesh with the format Gmsh</td>
</tr>
<tr>
<td>msup:</td>
<td>mesh with the format IDEAS</td>
</tr>
<tr>
<td>datg:</td>
<td>command file Gibi</td>
</tr>
<tr>
<td>pos:</td>
<td>results file with the format Gmsh</td>
</tr>
<tr>
<td>ensi:</td>
<td>directory result with ensight format</td>
</tr>
<tr>
<td>the dat:</td>
<td>results file containing of the curves to format XMGRACE</td>
</tr>
<tr>
<td>PS:</td>
<td>file with the format postscript</td>
</tr>
<tr>
<td>agraf:</td>
<td>results file containing the data for Agraf (the old versions of Aster wrote the directives and the given in the same file that it was necessary to cut out with the command post_agraf on the machine of reference)</td>
</tr>
<tr>
<td>digr:</td>
<td>results file containing the directives for Agraf</td>
</tr>
<tr>
<td>rmed:</td>
<td>results file with med format</td>
</tr>
<tr>
<td>the unv:</td>
<td>results file with the “Universal” format (IDEAS)</td>
</tr>
<tr>
<td>distr:</td>
<td>file of the values used for a parametric study</td>
</tr>
<tr>
<td>hostfile:</td>
<td>file describing the resources machines to use (parametric study)</td>
</tr>
<tr>
<td>name:</td>
<td>to recover from its name, a file present in the temporary directory /tmp</td>
</tr>
<tr>
<td>para:</td>
<td>file of parameters (retranscription of the parameters of computation for the tests)</td>
</tr>
<tr>
<td>repe:</td>
<td>result and/or data directory (makes it possible to transmit/recover the complete contents of a directory; as one does not affect numbers of logical unit Aster must reach the files by their names).</td>
</tr>
</tbody>
</table>
directory of the results during a parametric study
libr: file or directory with the choice of the user
btc: script of launching generated by the service (one can thus recover it, to modify it…).

During a astout on the machine of reference, \texttt{resu\_test} must be on this one.
During a parametric study, the directory of the results (standard \texttt{repe}) must be on the object computer.

2.2.2 Button REX

This button allows:

- **To emit a file without joining the profile**: This function makes it possible to the user to emit a file back from experiment to point out an anomaly in \texttt{Code\_Aster} (AL: anomaly software), to ask for an evolution of \texttt{Code\_Aster} (EL: software evolution), in one of the associated tools (Mongrel, Homard, Eficas, astk, bsf…) (AO: anomaly outil/EO: evolution tools), a modification of documentation (ED: documentation evolution), or a request for expert testimony in modelization (AOM: helps with the optimization of the modelization). The relative information with the transmitter of the file are accessible by small the Configuration/Interface. The files contained in the profile running are not joined to the file, which can make very difficult the processing of the file. It is advised to use "to emit a file and to associate the files with it". The version of \texttt{Code\_Aster} indicated is that selected in the interface (Parameters of computation).

- **To emit a file and to associate the files with it**: This function makes it possible to emit the same type of file as the preceding one, this time the data included in the profile are united with the file. 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!

- **To consult the files**: gives access the management tools back from experiment (on the machine of reference).

Notice

| features REX are available only on the server of reference. |
2.3 Parameters of submission

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

Figure 2.3-1: Zones of the parameters of computation

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
One defines the quantity of:
- Mémoire totale used for the job (in megabytes).
- Of which Aster: limit memory used by Aster (and thus memory for other products leaves).
- the maximum time of computation (in seconds, minutes: seconds or hour: time: seconds).

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

debug/nodebug: for a study without overload, one specifies which executable one wishes to use (provided both are available); during an overload, one chooses to compile with or without information of debug.

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 (with or without overload), 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 means of the débuguer,
- "pre": prepare the working directory 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 as of buttons of the interface. During the launching of a computation, the operations in progress are displayed here.

During the launching of a computation, the operations in progress are displayed here.

2.5 State by the jobs

This window provides information on the state of launched computations, which appear in the form of a list. While clicking with the right button on a computation, 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 output (stdout) and the output of (stderr) error. 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 directory then open with the local editor.

- **To download results files** the force the recopy of results files. This action is made automatically when computation passes in state ENDED. If this stage failed (problem network), that would make it possible to carry it out it again.

- **To bring up to date** questions the servers on whom computations are in progress. The check box makes it possible to reactualize automatically with a frequency defined in Configuration/Interface.

- **To bring up to date all** refreshes the state of all not finished computations.

- **To remove** erases the selected jobs of the list, the files related to this job in the flashor and stops computation if this one is not finished (by sending signal SIGKILL).

- **To stop properly** sends signal SIGUSR1 to computation. Code_Aster intercepts this signal to stop computation, 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 servers and removes there the files of the directory flashor nonaccessible since the follow-up from the jobs.

- **To search** makes it possible to consult the last lines of the message file 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 statement).

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

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)
- Goes back to submission
- Hour from State
- submission from the job (HANGS, RUN, SUSPENDED, ENDED)
- Diagnosis from job (OK, NOOK, <A>_ALARM, <F>_ERROR, <F>_ABNORMAL…)
- Name of the tail in batch or “interactive”
- TEMPS CPU of the execution Login
- Aster on the server of computation used
- Address of the server of computation used
- Machine of computation (name of the node for a cluster)
- Version of astk
- Indicating interactive batch/
3 Limps of selection of files: bsf

bsf is tools delivered with astk 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 servers.

bsf uses the configuration of the servers of astk, in particular the fields necessary to connection (addresses IP, login) and the commands 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 astk, it is thus necessary to close the bsf then to open it again.

Note:
bsf does not treat the directory and file names containing spaces (it results from it a shift in the names and the standard 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 server to another, the second memorizes the list of the last twelve directories 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 directory which one seizes the name. Generally, when the directory required does not exist, one turns over in the HOME defined for the current server.

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

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

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

Figure 3.1-1: Window of configuration of the remenant directories
3.2 Finely 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/directories (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/directories.
- **To re-elect**: give a new name to a file
- **New file/New directory**: respectively allow to create a file or a directory in the current directory.
- **To carry out a command**: give the possibility of carrying out a command 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 server.
- **Terminal**: open a terminal on the current server.

3.3 Small Display

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

bsf poster the directories 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 display 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 directory $HOME/.astkrc.

If the user wants to return to the configuration of origin, it must erase this directory and start again astk.

**Notice n°1**

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

**Notice n°2**

When astk carries out a command which tries to display windows: if the command is carried out locally, not problem. If it is carried out on a distant machine, it is to it by means of HS - X, therefore the display must occur well if server 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 server of reference: the complete name of this server and the login of the user. This last must be defined in this file to avoid following alarm:

<code>&lt;A&gt;_ALARM remote connection may fail: devel_server_user not defined in /home/xxxxxx/.astkrc/prefs</code>
4.1 Servers

One reaches the window of configuration by small the Configuration/Servers (see Figure 2.1.2-2).

The first button makes it possible to pass from a server to another and to add a “New server”. The fields are:

- Complete name or addresses IP: it is the name of the server on the network; one can indicate his complete name with domain name (for example: linux.lab.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 server
- : one can put on “off” a temporarily inaccessible server. Login: identifying
- with which one connects oneself to the server. Directory HOME
- : directory by default when one surfs on this machine. Directory of
- the services: directory where are installed the services on this server (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 server), automatic (astk questions the server with starting every 30 days). Last download
- : date from the last upgrade of information of configuration. Terminal: order
- to open a terminal on the server. This makes it possible to open a window of commands on the server 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 accessible nodes of computation since this server. Interactive batch/
- : specify if the server accepts launching in batch, interactive and provides the limits in memory, TEMPS CPU, many built-in processors on the server. The supported management software of batch is LSF, Sun Grid Engine and PROBLEMS. Configurations

4.2 General

4.2.1 Preferences

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

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 commands to reach a terminal and to an editor (as for the servers). Many profiles
in the small File make it possible to preserve the name of N the last open profiles so as to point out them quickly. Level of message

indicates the level of details of the messages written in the window of the messages of the menu Helps. 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 as of problems of communication with the distant servers. 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. Preferences

4.2.2 Network One reaches

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
the window of configuration of the preferences concerning the network by small the Configuration Preferences/Network (see Figure 2.1.2-4 Figure 2.1.2-4

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, an alarm message 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 local machine the IP address. While clicking on OK, the interface proposes IP addresses of the interfaces networks detected on the machine. Tools See Figure

4.3 2.1

.3-1 Figure 2.1.3-1

the tools to be configured with the drop-down list, or new tools 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 in the mitre STUDY, or on a file when one traverses the filesystem with the bsf. One defines simply the command line necessary to the launching of tools (advised absolute path), one can place the codes @F, @R, @f, @D in the command line (see § 2.1.3) to correctly pass a file to the tools. One can put these codes between brackets to be able to launch the tools alone, without file in argument. Types of files can be associated with the tools. The type taken into account to launch the tools is the extension when one browses the files with the bsf, the type of the drop-down list when it is about a mitre. One can choose if the tools are 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 directory, of launching the tools, then to redeposit the file on the distant server (even if it were not modified by the tools). How to make
5  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 response in tension of a plate perforated by adapting the mesh. One has the following elements: The file of

- Aster command: demo001 a.comm  the description of
- the geometry carried out with Gmsh: demo001.datg The mesh coarse
- of the plate builds by Gmsh: demo001a.msh The mesh fine of the contour
- of the plate: demo001a.18 One produces the following

results: An Aster mesh file

- : demo001a.mail a results file with
- the format Gmsh (fields of displacements, error...): demo001a.pos a curve with the format png  
- produced by xmgrace): demo001a.png classical files of message
- and result Aster: demo001a.mess and demo001a.resu In the example, one places all
the files in the /home/tutorial directory. The files of this example are available in the directory astest of your version of Code_Aster , for example in /opt/aster/STA10.1/astest .

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. Creation of the profile One launches the interface

5.1  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.
Selection of the files Definition

5.2  of a basic path In

5.2.1  the mitre STUDY, one chooses

a basic path to simplify the access to the files. One clicks on the icon, one chooses
the /home/tutorial directory  . Addition of existing files One adds

5.2.2  the command file while

clicking on, the selection of file opens directly in the basic path which one has just defined. It only remains to select the file demo001a.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 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 mesh file

with the format Gmsh (demo001a.msh). astk recognizes the extension “msh”, the logical number of unit is positioned with 19, the box “D” is notched. In the same way for the mesh of contour demonstration
001a.18, astk identifies the type “libr” and positions the logical number of unit to 18. The box should be notched “D” because it is about a data file. One can also add the file demonstration 001a.datg. One strips the box “D”, it will not be used in the study but one can visualize this mesh by opening it with Gmsh (see §5.5). Addition of files… Except if an execution5.5

5.2.3 already took place,
results files do not exist yet, one cannot thus add them by traversing the tree structure. … while inserting blank line The mesh into

5.2.3.1 the Gmsh format will be read again and converted
in the file of Aster commands by the command PRE_GMSH into mesh with the Aster format. One can recover this mesh by adding a file of type “mail” on the logical unit 20. One clicks on, line is added in the list. One chooses standard “the mail” in the list (what causes to position the logical number of unit to 20). One indicates the name /home/tutorial/demo001a.mail or demo001a.mail or ./demo001a.mail (since one can indicate the name of the file into relative compared to the basic path). The file is produced by the execution, one thus notches the box “R” (result) and one strips “D”. Notice command PRE_GMSH uses by default
numbers

19 and 20 with the inputs/outputs, if one modifies the command file to read again or write the mesh files on other units, it is necessary to be coherent with the numbers indicated in astk. ... with “Value by default” One could

5.2.3.2 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 contextual), one will thus record 2.2.1. One chooses To record under… in the small File
5.2.1, one goes with the browser in the /home/tutorial directory, and in line the Selection, one types demo001 has (the extension. astk is automatically added). Let us note that the title of the principal window D
5.2.1 “astk gives the name of the current profile. The title is now: ASTK version 1.8.0 - demo001a.astk -
/home/tutorial One inserts blank line while clicking on, one chooses
the type of file “pos”, 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 path (see [§5.2.1]), name of profile (by withdrawing L”extension5.2.1 of the type “pos”, that is to say /home/tutorial/demo001a.pos. One sees as follows: ./demo001a.pos (name relating to the basic path). The box “R” was notched, and the number D
5.2.1 “unit logical fixed at 37. In the command file, one indicated: IMPR_RESU (UNITE=51,...) one thus modifies the number
D” logical unit consequently
5.2.1, it is enough to click on the old value, to erase it and type 51. 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. One proceeds of the same way to add demonstration

001a.png: to select the type “libr”, to modify the logical unit into 53 and the noun of the file. In the same way, one adds a file of type “mess” and

one of type “resu” of this way (to leave the numbers of logical unit by default). Appear 5.2.3.2 - 1: Window of the profile study To remove

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
5.2.4 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! Launching of computation the files given and results are selected, one adjusts the parameters of computation (see § 2.3), and one clicks on the button “Throw”. One must not forget that 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 computation 5.2.3.2 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 computation. The initial state of computation east HANGS (on standby), when computation starts, it becomes RUN, then ENDED when he is finished (other states are possible in batch). The button “To bring up to date” calls the service which refreshes the state of computations in progress. When computation is finished, one can consult the output of the job while double-clicking on the job, or by Publishing/File output. Figure 5.3-1: Window of follow-up of the jobs Consultation

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
while double-clicking on their name, which opens a text editor for the files “mess” and “resu”; on the results file with the Gmsh format, “pos”, that cause to open this file in Gmsh directly. One visualizes the deformed shape thus and the isovaleurs (provided Gmsh were installed, and that “pos” is in the types of files associated with Gmsh, to see [§ 4.3]). NB: The directory having to accommodate a results file does not exist, it 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 directory from the object computer. An alarm <A>_COPY_RESULTS shows the way where it is necessary to go to seek the results. 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. In this example, demo001a.datg is a file that Code_Aster cannot read again; it contains the description of the geometry, Gmsh uses it to create the mesh (.msh). One can nevertheless insert it in the profile (button ), to affect an unspecified type to him (“libr” for example) since it will not be used during the execution (boxes D, R not notched). One can directly open the geometry while making Open with…/Gmsh (right click on the name of the file), modify the geometry or the parameters of the mesh, re-mesh and record the mesh. One can then start again computation on the new file demo001a.msh. Obviously, this is not restricted in Gmsh; one can use other tools (mesh generators, tools of post-processing, graph plotter…) directly since astk and to reach thus all the files of a study since a profile with the adequate tools. Advanced features exectool One chooses in astk

### 5.4 simply

5.4 simply

5.5 the version of Code_Aster,
5.6.1 the debug mode

or nodebug, a possible overload and this will result in using such or such executable Code_Aster. One can be even further specifying the way exact to launch this executable: 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 argument 1 argument2... To use the option exectool comes down to launching: cmde_exec ./aster.exe argument1 argument2... In small the Options, one can directly specify cmde_exec

or a name of tools 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 command time accepts exactly this kind of argument (executable and its arguments ), one will thus have the execution time of computation. That does not have great interest, Code_Aster gives 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 command valgrind... which 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. How to launch a parametric study One understands by parametric study
6 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. Computations are launched in parallel: to see §7.1 concerning the specific options. Launching of computations in parallel§7.1 Distribution of computations the management

7 of distributed computations is activated

7.1 during the launching

of a parametric study. Indeed, each computation is independent of the different one. One can thus subject them in parallel to reduce the time of return. Use of the available resources One can insert in the profile

7.1.1 (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 computations could be subjected on compute

01 (insofar as they do not ask more than 8192 Mo them 4 ) and 1 computation on compute02 using less than 1024 Mo. In batch, one can subject much more computations than of processors available and leave the manager batch distribute computations on a cluster for example. In this case, one can fix cpu=50 to leave to the maximum 50 computations 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 mémoire totale are automatically given. Remarks One can easily collapse a machine while launching too many computations

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
with respect to the available resources. It is advised to get information about the use potential of layers of shared computations (class batch dedicated for example). Parallel computations count well for the number of processors which they use and not for 1. Before the launching of computations, connection with the nodes of computation is tested. The list will be restricted with the machines having been contacted successfully. Time of expiry When the number of computations to launching is very high.

7.1.2 higher than number

of processors overall available (and it is often the case), of computations are on standby of submission. If a computation 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 at the time of master computation, i.e. the time chosen in astk during the total submission. If no computation were subjected during this time, computation is rejected. Activation of the parallelism of Code_Aster internal parallelism

7.2 with Code_Aster is present in two forms

: Parallelism OpenMP functions in shared memory and is available
  • only in solver MULT_FRONT. 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 in the MUMPS solver and elementary computations. 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 computations) in small the Options (see §2.1.4). Multiple executions It is about a mode of quite particular

7.3 bound for

the developers. The purpose is simultaneously to carry out a profile (a study or a list of benchmark) on several machines. One activates it by notching multiple=oui in small the 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. The results, including the output files output and error usually copied in the directory flashor, are copied in directory $HOME/MULTI ($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 computation compatible with the resources of each machine, etc Use of as_run When L” one uses the interface astk for
8 launching computations

, this one as a customer calls on proposed services by a server who can be on the same machine or a distant machine (if the server is distant, there are file-swappings and a Shell command through the network (protocol rsh or HS) that we do not detail here). The features of as_run can be classified in several
categories: for the user: features which can be called

- by the users, such as for example launching Code_Aster “to 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 catalog of messages... reserved to the customers: irrelevant features direct
- out of astk (or another customer), for the tasks of administration: restrain towards the tools of follow-up of
- the anomalies, data base d'études... The 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 environment variables (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 For
the user - Lance execution
 described by the profile (action

8.1 by default)

: as_run --run [options] nom_profil.export The export file 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
- line of commands of Code_Aster
- , F: definition of a file, R: definition of a directory , N
- : used only for the workshop
- of software engineering (AGLA).
- The format for the parameters and arguments is: P nom_parametre

value or A nom_argument value. For the files and directories, the format is: Standard
F/R unite_logic
path DRC where DRC specifies if the file or
directory is in Data, Result
( both are possible group), and if the contents Are compressed. - A computation starting from the
files in arguments Carries out quickly

: 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). - 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 config.txt file, of the version by default or chosen 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 key words of Code_Aster. For the developer Notices These functions are not maintained and

8.2 could more 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 a command Code_Aster: as_

run --showop [options] command [.capy] - Turns over the numbers available for the routines YOU, OP, LC...

Turn over the first 8 results except if --all is present. as_run --free_sub [--all] - Built a file export to launch a benchmark and prints it on stdout: as_run --get_export testcase_name - Built a list of benchmark from commands/key words 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 Code_Aster benchmarks (since DIRi or the directory astest by default) and a file pickled writes result: as_run --diag [--astest_dir=DIR1, [DIR2]] [--test_liste=LISTE] [--only_nook] [diag_result.pick] - Operation on the catalogs 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 catalogs and displays some statistics. move = moves a message of a catalog with another and produces the modified catalogs and source files. as_run --messages subroutine | message_number as_run --messages check [--fort=...] [--python=...] [--unigest=...] as_run --messages move oldmsgid newmsgid - Turns over information on the processors and the memory of the machines

given: as_run --get_informations [--output=FILE] host1 [host2 [...] to maintain an installation local Notices These features

8.3 are not more usable with the current versions

|of Code_Aster. - The number of the version of development Turns over: as_run --getversion

[options] - Turns over the path of installation of the version of development:

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
as_run --getversion_path [options] - Built a version of Code_Aster
(executable, libraries, catalogs)

). “target `can be all or clean : as_run --make [-vers=VERS] [target] as_run --make [-vers=VERS] clean
Example: as_run --make clean bibc/
hdf removes the files objects associated
with the source files with the directory bibc/hdf. - Carries out one or more update of a version of development

(to prefer --auto_update which is given the responsibility to download all the updates and to apply
them in the good order): as_run --update [options] fich1.tar.gz [fich2.tar.gz...] - And the Downloads
the updates available since a server applies
to a version of development: as_run --auto_update [--vers=... [--force_upgrade] [--keep_increment] [--
report_to=...] The option keep_increment announces that
all the intermediate versions

are preserved (executable, catalogs, moduli python). The option force_upgrade makes it
possible to pass for example from version 10.0.30
to version 10.1.1 (what is licit because the 10.0.30 was stabilized under number 10.1.0). The
option report_to sends a mail summarizing the update (success or failure).
- Downloads the updates available since a server applies

astk/asrun itself: as_run --astk_update [--report_to=...] [version] version makes it possible to install
a particular version of astk (value of type 1.8.0). NB: astk_update is called before each update by auto_update.
- Construction of the ctags: as_run --ctags [--vers=VERS] For the interfaces

launching of computations - Lance
an execution (by calling

8.4 as_run in a separate process):

as_run --serv user@mach: /nom_profil.export - Sends the contents of “filename”
(possibly on a distant machine)
to the addresses email1, email2,...: as_run --sendmail [--report_to=EMAIL1, EMAIL2] filename - Turns
over information
on the configuration of a server: batch, interactive
(yes/not, limits), nodes of computation, versions: as_run --information - Turns over the state, the
diagnosis, the node of computation, time
CPU and the working directory
of a computation: 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: ace
_run --tail job_number job_name mode fdest nb_lines [regexp] - Stops
a computation and destroys the associated files: as_run --del
job_number
job_name mode [node] [--signal=...] - Removes the files
of computations which are not in the list: as_run --

purge-flash job_number1 [job_number2 [...] ] - Converts a file
(possibly distant) with format HTML and writes

result in FILE: in beta release as_run --convert-to-html [user@machine:] file --output=FILE

For the tasks
of administration - Inserts a new entry in the system

8.5 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 “corrvdev
” or “corrvexpl” (according to worms) in the files found

in “histor” and possibly the farm: as_run --close_issue --vers=VERS histor - Extract contents of the
files listed

in “input_file ‘in “histor ‘: ace

_run --extract_histor [--status=STAT] [--format=FORMES] [--all_msg]

input_file histor - Prepares the profile to insert an execution
in database:

as_run --insert-in-db [export_file] To supervise the launching and the return of
a computation One provides here methodology

8.6 for launching computations since a local machine

towards a server of computation, except ihm graphic, by calling the various services of as_run. The
explanations provided hereafter allow: to launch a computation, to supervise

a computation (to know its state), to recover
• the results, to publish
• the files output/error. Assumptions One
• explains the various stages
• by taking the example of the execution

8.6.1 of

a computation on the Aster5 server since a Calibre7 machine. The data files must necessarily be on
the station Gauges, that which subjects computation, or on the server of execution (.export, data files,
results, bases,...). In the examples, the following names of machines are used (and thus with adapting
to each situation): Calibre7 (customer): Aster5 (server): hostname=aster5.hpc.edf.fr, to use
• =NNI the heading used in the files
• .export to identify the server is (it is necessary to modify

Ni and): P P mcclient server aster5.hpc.edf.fr P username NOR P protocol_exec
asrun.plugins.server.SSHServer P protocol
_cpyto asrun.plugins.server.SCPServer
P protocol_copyfrom asrun.plugins.server.SCPServer
P aster_root
/home/rd-ap-simumeca P to proxy_dir /scratch/NNI
P preliminary platform LINUX64 Checking
: Connection in command line HS aster5.hpc.edf.fr pwd
must function without password
. Commands the addition of

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
the option --nodebug

stderr removes the printing of the output stderr of as_run. Recovery of information of the server (also allows to test

8.6.2 connection

) as_run --nodebug_stderr --proxy -- information info.export with info.export which does not contain the heading

8.6.2.1 . In particular, one can see the list of the versions available. Then,

the version testing will be used. Launching of a computation

ace_run --nodebug_stderr --proxy --serv study.export
with study.export:: <<< heading >>> + P actions make_etude P nomjob jobname P origin ASTK 1.13.4 P batch

8.6.2.2 mode P consbtc yes P

soumbtc yes P version testing <<< provided by --information

P tpsjob 10 A memjeveux

512 A tpmmax

600 P memjob 4194304
F COM $HOME/dev/ codeaster/src/astest/zzzz 100f.comm D1
F mess /tmp/zzzz100f.mess R
6 F resu zzzz100f.resu R 8 R bases NNI@aster5.hpc.edf.fr :
/home/NNI/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 directory 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 server (or will have to remain there as a result, case of the base in the example).

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: <<< heading >>> +

8.6.2.3 P jobid 12399454

<<< provided in the output of --serv P batch mode P nomjob jobname P tail_nbline 50 <<< only for --tail P edit_type output <<< only for --edict

Actualization of the state of the job ace_run --nodebug_stderr --proxy --news

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
8.6.2.4 it is necessary to indicate asrun.plugins.default.actualité_simple

 the diagram by default makes *actuality + get_

 results*. If one wants to make only actualization

8.6.2.5 of the output of a job in progress

 as_run --nodebug_stderr --proxy --tail post.export To publish

8.6.2.6 the file output

 as_run --nodebug_stderr --proxy --edict post.export possible Values for edict

 _type: output, error, export, script, diag

8.6.2.7 Options --version show program' S version number

 and exit - H, --help show this help message and

8.6.2.8 exit - v, --verbose makes

 the program more talkative --functions as silently

 as possible - G silent, --debug poster information of debug --

8.7 stdout=

 FILE makes it possible to redirect the messages usually written on sys.stdout --stderr=FILE makes it possible

to redirect the messages usually written on sys.stderr (only messages of asrun) --log

_progress=FILE redirects information of advance

towards a file instead of sys.stderr --nodebug_stderr removes the printing of information of debugging on stderr - F, --force force the operations which can use a mask (download, compilation...) --display =DISPLAY value of variable DISPLAY (NB: certain functions read

this value in a file) --rmdir=DIR uses the directory of resources

$HOME/"DIR" (the default is .astkrc). It is necessary to avoid showing the absolute way because it will have passed to the distant servers. --remote_shell_protocol=REMOTE_SHELL _PROTOCOL protocol used for the launching

of commands --remote_copy_protocol=REMOTE_COPY_PROTOCOL protocol used for the copy of files --copy_all_results

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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
Copy all results in the current directory (for --quick action) --proxy request with a server to launch the specified action (for example, by calling as_run --serv on the server) --schema=SCHEMA makes it possible to modify the behavior of as_run by means of an alternative diagram Options for the maintenance actions: --filter=FILTER filters applied to the parameters of the tests: "name will _para < value" (supported comparisons <, >, =). --vers=VERS Version of Code_Aster to be used (for get, show, showop) --force_upgrade Force the update towards the next version (for example of 10.1 .xx towards 10.2.0) --output=FILE redirects result in FILE instead of stdout --surch_pyt=REP one or more directories (separated by a comma) containing the added files python --configuration=FILE uses another file “config.txt” (only for make, update and auto_update). --only_nook brings back only the errors (the time spent in tests ok is counted) --surch_fort=REP one or more directories (separated by a comma) containing added files FORTRAN - has, --all copy of all the files of the benchmark --to destdir =DIR root directory (fake-root) where the files will be copied --keep_increment a version increment by increment by preserving the executable files intermediary updates --search=REGEXP preserves the tests checking the regular statement given (or a simple character string). --unigest=FILE a file unigest (for the suppression) --command=COMMANDE [/MOTCLEFACT [/MOTCLE [=VALEUR]]] preserves the tests and the using the command key words given --test_liste=FILE list of the benchmarks --report_to=EMAIL addresses transport where to send the ratio of an 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 directory of the benchmarks to be analyzed - L, --room the files are not sought on the distant server but on the local machine --nolocal force the search of the files on the distant server (opposite of --room) Options for operations one jobs: --signal=SIGNAL sends a signal to computation (KILL |USR1) --result_to_output Redirects result in FILE instead of stdout Options for the interface with the REX: --status=STAT an error occurs if the files are not in this state --all_msg recovers all the messages of the files --format=FORMES format of the histor generated (text or HTML ) additional Precise details When one recovers a source file with get, show or showop , this one is put in /tmp/astk_ `login ` /cache. If 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.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
for the same file again is asked, this one is taken directly in this directory except if L" option --force

8.7.1 is activated; the files

- which have more D" 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 default. In this case, one will use --nolocal to reach a version

- on the server of reference. During the maintenance of a local installation, the default is to download the updates on a distant server. 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, L" user can define certain values in $HOME/.astkrc/prefs (where $HOME is the directory by default of L" user), in particular the name D” user to use on

- the distant server (devel_server_user) to avoid an alarm with each launching of as_run. Complements on the inner working of ace_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/. Frequently asked questions

- “Nothing occurs when one tries to launch a computation, D” to publish a file or to open a terminal on a distant server” or “Return code = 2, Profiles Copy failed in the window of the messages to the launching of a computation”

9 It is probable that

**the local machine cannot communicate correctly with the distant server.** The communications using protocol CCP/rsh, the files .rhosts on the local machine and the distant servers must be correctly indicated.

If one uses protocol SCP/HS, it is necessary that the private and public words are coherent. One can check it by leaving astk, and by starting again it of this way : astk --debug 0 - - check (--debug 0 is used to limit the printings) To attentively read information which is printed. astk provides information on environment variable DISPLAY (to check that it is correctly defined), and tests connection between the machine where astk is launched and the distant servers. In the event of problem of connection with a server, astk suggests a modification of the file .rhosts of the target machine. “When a computation 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 works. 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 by providing the traces of execution. 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). To reproduce the problem
by making the minimum of thing. To leave astk and join to the support the file $HOME/.astkrc/log.astk (or
1. in Salome-Meca).
2.
3.