Organization (S): EDF/IMA/MMN, GIST

Data-processing Manuel de Descriptif
D1.02 booklet: Workshop of Génie Logiciel of *Aster*
Document D1.02.03

Specification of the AGLA

Summary:

This document describes the tools of the AGLA from a data-processing point of view. It gives the functional specifications which were used for the realization of these tools.

Contents

1 General information 7
2 Definition of the source code of Code Aster 8
3 Instrumentation of the source code 9
3.1 Code of the type FORTRAN 9
3.2 Code of the type CAL 11
3.3 Code of the type C 11
3.4 Code of the type CATALOGU 12
3.5 File of the type CASTEST 12
3.6 File of the type UNIGEST 14
3.7 Types of files 14
4 File of identification ident Aster 15
5 Files of the noted code (quirou, quicat and quites)

..... 15
6 Files lists of the CAS-tests 19
6.1 List of management of the CAS-tests (List) 19
6.2 List of all the CAS-tests (liste_ct.tout) 20
6.3 Short list of CAS-tests (liste_ct.rest) 21
7 File of authorization of the restitutions (lien_dvp) 22
8 Tools of notation of the source code 22
8.1 asno : notation of units 24
8.2 asdeno : denotation of units 26
8.3 xasdeno : denotation of units starting from their name 26
8.4 asdeno_adm : denotation by the administrator 27
8.5 asqui : list of notation of units 28
8.6 asquit : list of all the notations 28

9 Tools of checking and restitution 28
9.1 aslien : delegation of the right of restitution 29
9.2 aslibe : release of the right of restitution 30
9.3 asquil : list of the delegations of restitution 31
9.4 asverif : checking of source 31
9.5 asrest : restitution of source 36
9.6 as.tout : passage of CAS-tests 40
9.7 ccat92 : compilation of the catalogues 46
10 Tools of update of the version 48
10.1 majnew : update of the version NEW 48
10.2 finmaj : end of update of the version NEW 55
10.3 actulist : actualization of the cost of the various lists of CAS-tests 56
10.4 majlist : update of the complete listings and restricted

..... 57
10.5 majsta : stabilization of a version 58
10.6 Change of version NEW 60
11 Compilation and edition of the links 61
11.1 cft77_aster : official module of compilation ofAster for FORTRAN 61
11.2 cc_aster : official module of compilation ofAster for the language C 62
11.3 as_aster : official module of compilation ofAster for assembler CRAY (CAL) 63
11.4 segldr_aster: official module of edition of the links ofAster 63
12 Management tools of the AGLA by the administrator 65
12.1 bolt : locking of the AGLA for the developers 65
12.2 deverrou : unlocking of the AGLA 65
12.3 fgrep_agla : "fast-grep" in the sources of the AGLA 65
12.4 stat_agla.qsub : trees of static call of scripts-shells 65

12.5 app_agla.qsub : appealing of script-SHELL
.....
67
13 Tools for Aster
.....
67
13.1 Safeguard of the sources on IBM: limited to NEW2
.....
67
13.2 Safeguard of the sources on Sun cassette
.....
69
13.3 aclaut Access controls of files and repertoires
.....
69

1 General information

Rules of use of the tools described here to manage the software configuration of *Code_Aster* aim not too much not forcing the developers and at guaranteeing that the evolutions brought do not come to deteriorate the good performance of the developments already carried out.

These goals will be achieved only:

- if the developers do not seek to cheat with the system (the experiment showed us that the already practised rules led to a comfort such as no one did not try to make it),
- because the code is managed in a centralized way.

These rules are based mainly on three concepts:

- extraction of the source, the system is tiny room to its more simple expression since one is satisfied to inform the community which one wishes to modify part of the code given,
- restitution, one checks that there are no conflicts between the code returned by a developer, the code developed by his colleagues and the code already returned in the version of reference,
- systematic passage of a set of tests of reference before taking into account of the modifications.

The tools described in this document will be scripts in C-SHELL when that is possible and of the programs C for the rest.

For the moment, all these tools must be available on `CRAY-C 98`.

All the tools must be able to be used in batch (`qsub`) and in interactive.

Note of presentation:

- `nom_en_police_courier_minuscule` :
  - represent a file CRAY of the AGLA (with the direction general i.e. file or repertoire or order),
  - also represent an order or a concept related to CRAY and UNICOS,
- `nom_en_police_courier_minuscule_italic` :
  - represent a variable of a tool of the AGLA.

2 Definition of the source code of *Code Aster*

The source code of *Code Aster* is made up:

- routines FORTRAN stored by LIBRARYS. We will speak then about code FORTRAN.
- functions in language C.
- modules out of assembler CRAY: CRAY Assembly Language or CAL. The assembler is also gathered in libraries. Like C and it FORTRAN CAL is gathered in library.
- elements of CATALOGUES to some extent complementing the programming of certain parts of the code (Language process control, Finite elements). We will speak then about code CATALOGUE.

  Example:

```
      % MODIFICATION DMEDX TYPELEM     DATE 12/12/91
fgazzpe AUTHOR
      ELEMENT D_DEPL_R_DX   % DDL BINDS IN MECHANICS (CMP
DX)
      MESH  SEG3     001
      MAP         4
      MTEMPSR  = INST_R    E  1  IDEN  1 INST
      MDDLMUR  = DDLM_R    E  1  IDEN  1 A1
      MDDLIMR  = DDLI_R    E  1  IDEN  1 C
      MDDLIMF  = DDLI_F    E  1  IDEN  1 C
      … etc…
```

  These elements of CATALOGUES are stored by Catalogues.

  Definition:

  One calls unit of source or more simply unit a routine FORTRAN or a module CAL O U a function C or an element of CATALOGUE. Cbe *two concepts are called units because they constitute indeed units within the meaning of the management of the sources.*

A version of*Aster* consists of source code but also associated CAS-tests. Without being source code, a CAS-test is essential. The cas-tests are also units and are subjected to the same constraints of management as the sources.

Sources and CAS-tests of *Code_Aster* are stored in files corresponding to versions of reference. The need for being able to make corrections on the version delivered to the users and to be able to make evolve the version of development results indeed in having to consider two versions of reference. Currently, it is the versions NEW2 and NEW3.

3 Instrumentation of the source code

The units of source and CAS-test must have a line comment standardized which must appear one and only once in each unit of source.

This line is called line INFORMATION.

There exist two types of lines INFORMATION:

- the line ADDITION who is intended to inform the modules of management which the unit of source is new.
- the line MODIFICATION who is intended to inform the modules of management which the unit of source comes to modify a unit of already existing source.

The lines of this last type should not be modified in **no case** by the developer.

Their form differs for code FORTRAN, CAL, C, CATALOGUE or CASTEST.

3.1 Code of the type FORTRAN

Lines INFORMATION are form:

```
SUBROUTINE nomsub
------ information of SUBROUTINE nomsub not taken into account by the
modules
of management ------
C MODIFICATION nombib_1  DATE jj/mm/aa AUTHOR user_CRAY
nom_autor
------ information of SUBROUTINE nomsub not taken into account by the
modules
of management ------
END
INTEGER FUNCTION nomfun
------ information of FUNCTION nomfun not taken into account by the
modules
of management ------
C ADDITION nombib_2
------ information of FUNCTION nomfun not taken into account by the
modules
of management ------
END
```

Syntax of a line INFORMATION (the various fields are separated by at least 1 white):

- "C" in first column,
- "MODIFICATION" or "ADDITION",
- nombib_i : name of the library,
  if "MODIFICATION":
  - "DATE",

- `jj/mm/aa:` date managed at the time of the update of the version of reference,
- "AUTHOR",
- `user_CRAY` : to use on CRAY of the last person who modified the routine,
- `nom_autor` : bound to the preceding identifier. These two identifiers are also managed at the time of the update of the version of reference.

These are the fields which should not be modified by the developer.

3.2 Code of the type CAL

Lines INFORMATION are form:

When one wants to add a routine the line should be put INFORMATION following in the module (this line delimits the beginning of the module):

```
* nom_de_bibliotheque ADDITION nom_module
```

- * in first column,
- ADDITION (in capital letters obligatorily!),
- the name of the module,
- the name of a library CAL *Aster* where the module must be added (this library must already exist - cf organization Source *Aster*-).

Note:

A module which one wants to add should not already exist in Aster (either in FORTRAN maybe in C maybe in CAL).

Once the module added to *Code_Aster*, the line INFORMATION is managed and put up to date by the AGLA. It has the following form then:

```
* MODIFICATION nom_module NOM_BIB DATES jj/mm/aa AUTHOR
J2BHHMB C.MASSERET
```

It contains then:

- the name of the module,
- the name of the library CAL where the module is,
- the date of its last modification,
- to use CRAY and the name of the author of the last modification.

3.3 Code of the type C

Lines INFORMATION are form:

```
/* nom_de_bibliotheque ADDITION nom_fonction *
```

- /* in first column,
- ADDITION (in capital letters obligatorily!),
- the name of the function,
- the name of a library C *Aster* where the function must be added (this library must already exist - cf organization Source *Aster*-).

Once the function added to *Code Aster*, the line INFORMATION is managed and put up to date by the AGLA. It has the following form then:

```
/* MODIFICATION nom_fonc NOM_BIB DATES jj/mm/aa AUTHOR
J2BHHMB C.MASSERET *
```

It contains then:

- the name of the function,

- the name of the library C where the function is,
- the date of its last modification,
- to use CRAY and the name of the author of the last modification.

Note:

> *So that a function C can be called in FORTRAN it is necessary that its name is in* ***capital letters*** .

3.4 Code of the type CATALOGU

Lines INFORMATION are form:

```
      ------------- information not taken into account by the modules
of management ------
      %& MODIFICATION nom_catalogue_1 DATE jj/mm/aa user_CRAY
AUTHOR
         nom_autor
      ------------- information of the element of the catalogue
nom_catalogue_1
not taken into account by the modules of management ------
      %& ADDITION nom_catalogue_2
      ------------- information of the element of the catalogue
nom_catalogue_2
not taken into account by the modules of management ------
```

Syntax of a line INFORMATION (the various fields are separated by at least 1 white):

- "%" in first column,
- "MODIFICATION" or "ADDITION",
- & in second column,
- nom_catalogist: name of the catalogue,
  if "MODIFICATION":
  - "DATE",
  - jj/mm/aa: date managed at the time of the update of the version of reference,
  - "AUTHOR",
  - user_CRAY : to use on CRAY of the last person who modified the routine,
  - nom_autor : bound to the preceding identifier. These two identifiers are also managed at the time of the update of the version of reference.

One notes, on this example that the line INFORMATION play the part of delimitor **contrary** with the case of FORTRAN.

3.5 File of the type CASTEST

For the CAS-test two lines INFORMATION are obligatory ADDITION or MODIFICATION and TITLE.

Lines INFORMATION are form:

- for the modification of a CAS-test:

```
          ------------ information not taken into account by the
modules of management ------
          % MODIFICATION  DATE jj/mm/aa  AUTHOR user_CRAY
nom_autor
          % TITRATES explanation, in light, of the role of this CAS-test
          ------------ information not taken into account by the
modules of management ----
          BEGINNING (CODE: (NAME: 'nom_du_cas_test',
```

- for the addition of a CAS-test:

```
          ------------ information not taken into account by the
modules of management ------
          % ADDITION
          % TITRATES explanation, in light, of the role of this CAS-test
          ------------ information not taken into account by the
modules of management ----
          BEGINNING (CODE: (NAME: 'nom_du_cas_test',
```

It should be noted that:

- the name of the file must correspond to that of the CAS-test,
- there is one CAS-test by file,
- the line `INFORMATION 'TITRE'` is to be informed by the developer.

3.6 File of the type UNIGEST

A file of the type UNIGEST expressly does not contain a source code, but instructions concerning the units of source for the modules of restitution.

In this file one indicates the units of source or CAS-test which one wishes to move of a library with another or to remove. This kind of file contains lines of the following type:

- FORSUPPR nom_de_routine nom_de_bibli
  for the routines which one wishes to remove, by indicating their name and the library of membership,
- CATSUPPR nom_de_sous_catalogist nom_de_catalogist
  for the elements of catalogues which one wishes to remove, by indicating their name and catalogues it membership,
- TESSUPPR nom_de_cas_test
  for the CAS-tests which one wishes to remove, by indicating their name,
- FORDEPLA nom_de_routine bibli1 bibli2
  for the routines which one wishes to move of a library bibli1 towards a library bibli2.

It should be noted that for this kind of file:

- there is no concept of comment,
- so that there is no inconsistency there cannot be more than one order UNIGEST concerning the same unit,
- it is not for the moment envisaged of orders equivalent for the source C and CAL.

3.7 Types of files

File of the type will be called:

- FORTRAN a file containing of the source of the type FORTRAN,
- CAL a file containing of the source of the type assembler CRAY,
- C a file containing of the source of type language C,
- CATALOGU a file containing of the source of the type CATALOGU,
- CASTEST a file containing of the source of the type CASTEST,
- UNIGEST a file containing of the source of the type UNIGEST.

Remarks :

One cannot mix in the same file the various types (FORTRAN, CAL, C, CATALOGU, CASTEST, UNIGEST),

For the additions, the date and the name of the author are obviously not necessary in the line INFORMATION.

4 File of identification ident *Aster*

The identification of the developer is carried out by sound to use Cray. The file ident *Aster*, residing on CRAY, allows to make the correspondence with its name. The update of ident *Aster* is ensured by the administrator *of Code_Aster*. One to use Cray can appear only once in this file. To be able to use the tools of the AGLA it is obligatory that it to use calling is defined in this file.

Syntax of the file ident *Aster* (the fields are separated by at least a white):

- to use CRAY,
- name,
- address e-mail,
- department,
- group (for the statistics concerning the EDA),
- role in *Aster* among (one or more separated by one/):
  - EDA,
  - CUA,
  - ACUA,
  - YIELDED,
  - ASA,
  - CPA,
- type of machine for the interface asterix (if CUA) among:
  - SUN_OS,
  - SUN_SOLARIS,
  - HP,

Example:

```
%user CRAY              e-mail          Dpt
name
gjbhhts         Pascal.Mialon@der.ed  MN   I75  EDA/CEDA
P.MIALON        f.fr                  MN   I74  SUN_OS/HP
f6bhhbu D.BUI   Danièle.Bui.der.edf.          CUA/ACUA SUN
                fr
```

5 Files of the noted code (quirou, quicat and quites)

The tools of notation rest on:

- the file quirou residing on CRAY on which are registered:
  - names of the routines FORTRAN, modules CAL, and functions C noted (we will specify this concept thereafter) by the developers,
  - the name of the version of reference concerned or the keyword **RAYE** (if the unit were indicated by the administrator,
  - hh: mm: ss: the hour of notation,
  - jj/mm/aa: the date of notation,

- user_CRAY: the identifier of the developer on CRAY,
- the name of the developer,
- jj/mm/aa: the date of denotation by the administrator (when there is the keyword **RAYE** in the field version).

These files are sorted by date of increasing notation.

Example:

```
      -------------  ROUTINES CURRENTLY LEFT:
      TE0010   NEW1   17:32: 29   11/7/90   g8bhhxd
X.DESROCHES
      CHARGR   NEW2   17:00: 55   2/22/91   vabhht2   C.MENGONI
      CORDDL   NEW2   17:01: 21   2/22/91   vabhht2   C.MENGONI
      TE0040   NEW2   17:05: 30   2/4/91   b8bhhh1
D.SELIGMANN
      LEC151   RAYE   14:16: 55   3/20/91   c4gffhj   A.ZSCDR
3/25/91
      LEC151   NEW2   13:06: 43   4/24/91   d6bhhhm
A.M.DONORE
      LEC151   NEW2   15:26: 17   4/24/91   c4gffhj   A.ZSCDR
      LEC164   NEW2   15:10: 41   4/24/91   d6bhhhm
A.M.DONORE
      LEC747   NEW2   17:26: 13   4/24/91   d6bhhhm
A.M.DONORE
```

- the file `quicat` residing on Cray on which the same information as described above but concerning the elements of catalogue is registered of *Code Aster*.

  Example:

```
      ------------  CATALOGUES CURRENTLY LEFT:
      MELIE2DL   NEW2   08:51: 08   3/13/91   vabhhts
J.PELLET
      MELIE3DL   NEW2   08:51: 08   3/13/91   vabhhts
J.PELLET
      MELIE6DL   NEW2   08:51: 08   3/13/91   vabhhts
J.PELLET
      THLIE2D    NEW2   08:51: 08   3/13/91   vabhhts
J.PELLET
      THLIE3D    RAYE   08:59: 08   3/13/91   c4gffhj   A.ZSCDR
3/13/91
      THLIE3D    NEW2   09:47: 35   3/13/91   vabhhts
J.PELLET
      MECYSE3    NEW2   17:53: 39   5/29/91   gjbhhts
P.MIALON
```

- the file `quites` residing on Cray on which the same information as described above but concerning the CAS-tests is registered of *Code_Aster*.

  Example:

```
      ------------  CAS-TESTS CURRENTLY LEFT:
      ADLV100A   NEW2   08:51: 08   3/13/91   vabhhts
J.PELLET
      HPLV100A   NEW2   08:51: 08   3/13/91   vabhhts
J.PELLET
      HSNV100A   NEW2   08:51: 08   3/13/91   vabhhts
J.PELLET
```

```
      SDLL15A    NEW2    08:51: 08    3/13/91    vabhhts
J.PELLET
      TPLA01A    RAYE    08:59: 08    3/13/91    c4gffhj    A.ZSCDR
3/13/91
```

By definition, we will say that a unit of source or CAS-test is noted if its name is registered in one of the files of noted code (`quirou` or `quicat` or `quites`) and that it "is not striped".

Note:

The notations of more than three months will be removed files of notation during the use of `majnew`,

The notations "striped" of more than three months will be also removed during the use of `majnew`.

6 Files lists of the CAS-tests

These files are located in the repertoire `/aster/astest/vers` where `towards` indicate the associated version of development (`NEW3` for example).

6.1 List of management of the CAS-tests (`List`)

This list describes all the CAS-tests of validation of*Aster*. It makes it possible to classify each case - test among one of the categories following:

short list: the CAS-test will be subjected to validate a restitution (attribute '. '),
complete listing: the CAS-test is subjected only once per week by the administrator *Aster* (attribute 'S'),
list performance: the CAS-test is individually managed (attribute 'P').

This list is reactualized automatically by the update of a new version of*Aster* : `majnew`.

It understands the following fields (the separator being the character '|'):

- attribute of Addition 'With' or of Modification 'M' at the time of the last upgrade and '.' in the contrary case,
- type of list:
  '.'short list (standard of list by default when L'one adds a CAS-test),
  'S' complete listing,
  'P' list Performance,
- name of the CAS-test,
- total time of execution in seconds,
- memory requested in MW,
- a field comment to justify the ranking of the CAS-test in a kind of list (on 55 characters),
- the title of the CAS-test (begin obligatorily with the character '%').

At the end of the list and for each type it there a:

- the number of CAS-tests in the list,
- the cumulated time of execution of all the CAS-tests (in S),
- the cumulated cost of execution of all the CAS-tests in job of day, of night and weekend.

After an update of*Aster*, the administrator can modify the list by modifying the distribution of the CAS-tests in the various lists as well as the modification of the zone comment. This operation is carried out using a text editor.

The administrator has then the possibility of updating the costs of the various lists. From this list it can also update the short lists and total described below.

Example:

```
| . | P | sdnl102a | 00068 | 008 |                    | % TITRATES
Carried cab…
| . | . | ssll101b | 00009 | 008 |                    | % TITRATES
PROBLEM OF HP…
| . | S | ssll11c  | 00004 | 008 |                    | % TITRATES
lattice of B…
| With | . | ssll11d  | 00004 | 016 |                    | % TITRATES
Lattice of B…
| . | . | ssll12a  | 00011 | 008 |                    | % TITRATES
lattice of B…
| . | S | ssls101d | 00020 | 008 |    cf ssls100     | % TITRATES
Plate circul…
| M | S | sslv101g | 00031 | 016 |    Too much long        | %
TITRATES SSLV101/G    …
| M | . | sslv101h | 00028 | 016 | Affe_char_cine       | % TITRATES
SSLV101/H CTJ…
| . | S | ttll01e  | 00015 | 008 | cf tpll01          | % Shock
thermal on…
| . | . | ttnl02a  | 00115 | 008 |    Too much expensive cf.    | %
TITRATES transitory T…
|   |   |          |       |     |    hsnv102a          |
| . | . | zzzz100a | 00014 | 008 | Test plantings  | % TITRATES
functions, Na…
                              possible
#################################################################
#################                  NB CAS-tests   time (S)
day (F)   night (F)   weekend (F)
# Short list    0007         000265     000589      000228
000128
# Complete listing      0013         000512     001338
000455      000669
# Lists Performance   0001         000068     000151
000060      000034
#################################################################
################
```

6.2 List of all the CAS-tests (`liste_ct.tout`)

This list contains the name and the description of all the official CAS-tests of a version. It makes it possible to pass all the CAS-tests to each new version. This list is put up to date starting from the list of management of the CAS-tests described above. The cas-tests of performances are inserted in this list but are put in comment.

Any line beginning by '`%`' is a comment.

The title of the CAS-test begins obligatorily by '`%`'.

Classified alphabetically.

Example:

```
   %  List of all the official CAS-tests of version NEW3
      adlv100a% Piston coupled to a column of fluid
      ahlv100b% Guides wave at anechoic exit (plane waves)
      hplv100a% Parallelepiped. YOUNG modulus function of the
temperature.
      hsnv10a  % Thermoplasticity in simple traction
      mtlp100a% Soaks of an infinite bar with square section
   %  sdnl102a% TITRATES Carried cables three-phase with frames
and descents
```

6.3 Short list of CAS-tests (`liste_ct.rest`)

This list, of the same form than the preceding one, contains only the most relevant CAS-tests within the meaning of nonthe regression of the code. It makes it possible to pass the CAS-tests defined during each restitution of code by a developer. This list is put up to date by the administrator *Aster* starting from the list of management of the CAS-tests.

7 File of authorization of the restitutions (lien_dvp)

These files are located in the repertoire /aster/astest/vers where towards indicate the associated version.

The AGLA makes it possible to take into account the restitution grouped for the dependent developments. By default each developer is only entitled to return its developments. When developments are bound, one of the developers must centralize the restitutions. The other developers will have to then delegate their right to him to return their developments. As long as the recipient of this delegation (or the administrator) will not have given again their rights to the other developers, they will not be able to make restitution.

With each new version the rights are reset (each developer is then only the ability to return its developments).

The file lien_dvp contains only the delegations of restitution (not the rights by default). It is composed of four fields per recording:

- to use CRAY developer délégant its right of restitution,
- to use CRAY of the recipient of this delegation,
- version concerned with the delegation (NEW1 or NEW2 or NEW3),
- jj/mm/aa date of the delegation,
- (nom_developpeur_appelant - > nom_developpeur_beneficiaire) given starting from the file ident.

Example:

```
g8bhhhh  j2bhhmb  NEW3  8/5/92   (R.Michel - > C.Masseret)
```

For reasons of coherence, before the construction of the new version,

- to use CRAY must exist in the file ident,
- a developer having profited from at least a delegation will not be able to delegate her own right of restitution to another developer,
- a developer can delegate only once its right of restitution.

8 Tools of notation of the source code

These tools must be used by any person wishing to carry out a development in a version of reference of *Code_Aster*.

They are located in the repertoire /aster/astest/vers where towards indicate the associated version and are also available via the environment of use asterix [D1.02.01 §12].

Generally all the tools described thereafter return an error code via the code back from a process UNIX (consultable by $status in C-SHELL and $? in Bourne-SHELL).

This code is the following:

- 0 if all is OK,

- 2 if a message of alarm were transmitted,
- 4 if an error were met.

8.1 asno : notation of units

To note in the files of code noted (quirou or quicat or quites) units of source.

8.1.1 Mode of call

```
asno mon_fichier [version [ message ]]
```

mon_fichier indicate a file being is of the type FORTRAN or CAL or C or CATALOGUE or UNIGEST or CASTEST.

version is a parameter which is called after the version of reference concerned (NEW1, NEW2, NEW3). The value currently taken by default is NEW3.

message indicate an output file where the messages of alarms and errors emitted will be registered by asno. The name taken by default is: message_asno.

8.1.2 Tasks carried out

Three principal tasks:

With Research of the type of the file source code FORTRAN or CAL or C, source code CATALOGUE, UNIGEST, or CASTEST on the following criteria:

- If the file contains a line starting with:
  C ADDITION or C MODIFICATION
  C being in first column, the number of white enters C and ADDITION or enters C and MODIFICATION not being significant,
  then the file will be regarded as container of the code FORTRAN
- If the file contains a line starting with:
  /* ADDITION or/* MODIFICATION
  /* being in first column, the number of white enters /* and ADDITION or enters /* and MODIFICATION not being significant,
  then the file will be regarded as container of the code C
- If the file contains a line starting with:
  * ADDITION or * MODIFICATION
  l'* being in first column, the number of white between * and ADDITION or between * and MODIFICATION not being significant,
  then the file will be regarded as container of the code CAL
- If the file contains a line starting with:
  % ADDITION or % MODIFICATION
  % being in first column, the number of white enters % and ADDITION or enters % and MODIFICATION not being significant,

then the file will be regarded as container of the code CATALOGUE

- If the file contains a line starting with:

  % TITRATES

  % being in first column, the number of white enters % and TITLE not being significant,

  and a line beginning by:
  BEGINNING (CODE: (NAME: 'nom_cas_test'

  then the file will be regarded as container of the code CASTEST

- If the file contains a line starting with:

  FORSUPPR, CATSUPPR, TESSUPPR, FORDEPLA, then, the file will be considered type UNIGEST.

If the file contains only one type of source *Aster* :

the notation can continue

If not

an error message is registered and END.

B Scan for the noun of the units of sources and the name of the libraries or catalogues:

- In the case of a file of the type FORTRAN:

  The name of the routines is found according to the standards of declarations FORTRAN:

```
PROGRAM nom_unite
SUBROUTINE nom_unite
FUNCTION nom_unite
INTEGER FUNCTION nom_unite
… etc…
```

  The name of the libraries is found by the line INFORMATION.

- In the case of a file of the type CATALOGUE or CAL or C :

  The names of the units and the catalogue (or the library) are found thanks to the lines INFORMATION:

```
%&     MODIFICATION nom_catalogist DATES jj/mm/aa AUTHOR
nom_autor
```

  or

```
%&     ADDITION nom_catalogist
```

  % being in column 1 & in 2ème column and the number of white not being significant. (respectively with /* for C and * for CAL).

- In the case of a file of the type CASTEST :

  The names of the CAS-tests are found thanks to the name of the CAS-test in the instruction BEGINNING (CODE: (NAME: 'nom_cas_test',

- In the case of a file of the type UNIGEST :

  The names of the elements of unit are found in an obvious way.

C For each unit to be noted:

- If it is already noted in a file:

  If the first notation of this unit is the fact of the appealing developer asno:
  there is nothing has to make.

  If not:

  an alarm is emitted.

  The name of the people having noted this unit as well as the hours and dates of notation are registered in the file *message*.

  If the appealing developer asno does not have already noted this unit:
  the unit is noted

  If not:

  the unit of source is noted in the file of code noted with the appealing identifier, the hour and the date.

8.2 asdeno : denotation of units

To indicate units of source starting from a source file.

8.2.1 Mode of call

asdeno *mon_fichier* [*version* [*message*]]

mon_fichier indicate a file containing is code FORTRAN or CAL or C, that is to say code CATALOGUE, maybe of type UNIGEST or CASTEST.

version is a parameter which is called after the version of reference concerned (NEW1, NEW2, NEW3). The value currently taken by default is NEW3.

message indicate an output file where the messages of alarms and errors emitted will be registered by asdeno. The name taken by default is: message_asdeno.

8.2.2 Tasks carried out

- Task A identical to asno.
- Task B identical to asno.
- Research, in the file of noted code, of the units of source already noted by the appealing identifier, and removal of the corresponding lines.

8.3 xasdeno : denotation of units starting from their name

To indicate units of source starting from a file containing of the names of module and a kind of module.

8.3.1 Mode of call

```
xasdeno mon_fichier type [version [ message ]]
```

mon_fichier indicate a file containing a name of module per line the names of module must refer to units of the same type.

version is a parameter which is called after the version of reference concerned (NEW1, NEW2, NEW3). The value currently taken by default is NEW3.

type : to indicate the type of unit to be indicated among F (FORTRAN), C, cal(assembler), catastrophes (catalogues), test (CAS-test).

message indicate an output file where the messages of alarms and errors emitted will be registered by xasdeno. The name taken by default is: message_xasdeno.

8.3.2 Tasks carried out

Suppression, in the file of noted code, of the units of source already noted by the appealing identifier with the version given of the modules whose name is in the file.

8.4 asdeno_adm : denotation by the administrator

Possibility only given to the administrator *Aster* to invalidate all the notations concerning of the units. The 'indicated' units will have their field version replaced by the mention 'RAYE' and goes back it to this denotation will be added in the files of notation (so that the developers having noted these units are prevented).

The striped units will be 'visible' only with asqui and asquit.

8.4.1 Mode of call

```
asdeno_adm mon_fichier [version [ message ]]
```

mon_fichier indicate a file containing is code FORTRAN or CAL or C, that is to say code CATALOGUE, maybe of the type UNIGEST or CASTEST.

version is a parameter which is called after the version of reference concerned (NEW1, NEW2, NEW3). The value currently taken by default is NEW3.

message indicate an output file where the messages of alarms and errors emitted will be registered by asdeno_adm. The name taken by default is: message_asdeno_adm.

8.4.2 Tasks carried out

- Task A identical to asno
- Task B identical to asno
- Research, in the file of noted code, of the units of source already noted, replacement of the field version by 'RAYE' and addition of the date of denotation.

8.5 asqui : list of notation of units

To know the developers which noted units of source.

8.5.1 Mode of call

```
asqui mon_fichier  [ message ]
```

mon_fichier indicate a file containing is code FORTRAN or CAL or C, that is to say code CATALOGUE, maybe of the type UNIGEST or CASTEST.

message indicate an output file where the messages of alarms and errors emitted will be registered by asqui. The name taken by default is: message_asqui.

8.5.2 Tasks carried out

- Task A identical to asno.
- Task B identical to asno.
- Research in the file of noted code if the units of source are already noted (all versions are concerned).
  If such is the case:
  - The name of the units of source and the people as well as the hours and dates of notation are registered in *message* and END.
    (task identical to the task C of asno).

8.6 asquit : list of all the notations

To know all the noted units of source.

8.6.1 Mode of call

```
asquit [ message ]
```

message : identical definition with asno. The name taken by default is: message_asquit.

8.6.2 Task carried out

Recopy of the files of noted code (quirou and quicat and quites) in the file *message*.

9 Tools of checking and restitution

They are located in the repertoire /aster/astest/vers where towards indicate the associated version and are also available via the environment of asterix use [§12 D1.02.01].

9.1 aslien : delegation of the right of restitution

When developments are dependent it is necessary to be able simultaneously to return them in order to test the code. That obliges one of the developers to centralize the developments and to carry out the restitution for the other developers.

To avoid the problems, the developers must delegate their rights of restitution to that which centralizes the developments.

This delegation is exclusive for only one version and valid until the update of the new version. The delegation can however be returned by the recipient, or the administrator.

A developer can delegate only once this right (for a given version). The recipient of a delegation cannot delegate to him even his right to restore.

9.1.1 Mode of call

aslien *user_beneficiaire* [*version* [*message*]]

user_beneficiaire : to use CRAY developer profiting from the delegation of restitution.

version : is a parameter which is called after the version of reference concerned (NEW1, NEW2, NEW3). By default currently: NEW3.

message : definition identical to asno. By default: message_aslien.

9.1.2 Tasks carried out

To check that it *user_beneficiaire* exist in ident and is different from the appealing one (*iret* = 4).

To check that calling it did not already delegate its right of restitution (*iret* = 4).

To check that it *user_beneficiaire* did not already delegate its right of restitution (*iret* = 4).

To check that calling it is not already profit rights of restitution (*iret* = 4).

If *iret* < 4 to supplement the file lien_dvp with:

- to use the appealing one,
- user_beneficiaire,
- version,
- jj/mm/aa date of use of aslien,
- (nom_developpeur_appelant - > nom_developpeur_beneficiaire) given starting from the file ident.

9.2 aslibe : release of the right of restitution

Release of the delegation of restitution by the recipient or the administrator.

9.2.1 Mode of call

```
aslibe to use [ version [ message ] ]
```

to use : to use CRAY developer which delegated its right of restitution.

version : is a parameter which is called after the version of reference concerned (NEW1, NEW2, NEW3). By default currently: NEW3.

message : definition identical to asno. By default: message_aslibe.

9.2.2 Tasks carried out

To check that calling it and *to use* are different and defined in ident (*iret* = 4).

If calling it then removal of the line is the administrator of lien_dvp containing in first field *to use* for the version *version* (if it does not exist *iret* = 4).

If calling it then removal of the line is a developer of lien_dvp containing in first field *to use* and in second field it to use the appealing one for the version *version* (if it does not exist *iret* = 4).

9.3 asquil : list of the delegations of restitution

List in the file of message of asquil of all the granted delegations of restitution.

9.3.1 Mode of call

```
asquil [ message ]
```

message : definition identical to asno. By default: message_asquil.

9.3.2 Tasks carried out

To recopy the file lien_dvp in *message*.

9.4 asverif : checking of source

To check that a set of source can be introduced into a version of reference of *Code Aster*.

This tool carries out part of the tasks carried out by the module of restitution of source asrest who, will have, being used obligatorily by any person wishing indeed to restore a development.

asverif allows to check a grouped restitution, if the restitutions of several developers are dependent.

asverif work in a temporary repertoire.

9.4.1 Mode of call

```
asverif repertoire [version [ message ]]
```

repertoire is the name of a repertoire which contains a repertoire by developer wishing to restore source. The names of these repertoires must be it Use-CRAY developer to determine the author of the source restored. In each one of these repertoires one can find, according to the cases, the files:

repertoire/to use (S)/

histor	in which one finds a description of the evolutions brought (obligatory).
FORTRAN	in which source is FORTRAN.
cal	in which is assembler source.
catalog	in which is elements of catalogues.
u	
unigest	in which one indicates the units of source that one wishes to move of a library with another or to remove.
test	repertoire in which command files of CAS-tests are *.comm and possibly, of the files *.mail or will *.para

C repertoire in which files of the functions are C *.c.

message : definition identical to asno. By default: message_asverif.

version is a parameter which is called after the version of reference concerned (NEW1, NEW2, NEW3). By default currently: NEW3.

9.4.2 Tasks carried out

The tasks described below are it for all the files of the repertoires `to use (S)` contents in *`repertoire`*. These repertoires have the name of one `to use` CRAY contained in the file `ident` and for which the appealing one holds a delegation of restitution (file `lien_dvp`).

Only (S) it (S) file (S) `histor` is (are) obligatory (S), the actions described below, of course, are not carried out that if the files to which they refer exist. In certain case, actions of substitution will be carried out, they will be clearly specified.

- Destruction of all the repertoires and all the files being in (S) the repertoire (S) "`$TMPDIR/`*`to use (S)`*". Recopy of the contents of *`repertoire/to use (S)`* in the repertoires "`$TMPDIR/`*`to use (S)`*".

  The module works then starting from the files which are in `$TMPDIR/user (S)`.

  Repertoires `$TMPDIR/user (S)`*" are protected in writing with any user except with the user administrator*. `asverif` *(which has priorities of the administrator) works then only in the files of* `$TMPDIR/ser (S)`.
- Checking of the type of the files `FORTRAN`, `catalogu`, `unigest`, `test/*.comm` according to the criteria given in task 1 of `asno`.

  Checking that `histor` is nonempty (for each *`to use`*).
- Checks of a syntactic nature
  - For FORTRAN and histor:

    Detection of the lines of more than 72 characters and inscription of the corresponding lines in *`message`* (*`iret = 4`*).

    Detection of the lines and inscription of the corresponding lines in *`message`* (*`iret = 2`*) container:

    tiny characters,
    characters other than the alphabet A-Z, figures 0-9, space, the tabulation and the signs: `, ) (* + - =/ ' & § ! % : " . ? $ < > _`

    Detection of the lines and inscription of the corresponding lines in *`message`* (*`iret = 4`*) container: characters other than the alphabet A-Z and a-z, figures 0-9, space, the tabulation and the signs: `, ) (* + - =/ ' & § ! % : " . ? $ < >; é è # \ []`
  - For catalogu:

    Detection of the lines of more than 72 characters and inscription of the corresponding lines in *`message`* (*`iret = 4`*).

    Detection of the lines and inscription of the corresponding lines in *`message`* (*`iret = 4`*) container:

    characters other than the alphabet A-Z, figures 0-9 and the signs:
    `, ) (* + - =/ ' & § ! %; é è with _`
  - For `unigest` :

    Detection of the lines of more than 72 characters and inscription of the corresponding lines in *`message`* (*`iret = 4`*).

Checking which each unit is referred only one time on the whole of the developers.

- For test/*.comm:

  Detection of the lines of more than 72 characters and inscription of the corresponding lines in *message* (*iret* = 4).

  Detection of the lines and inscription of the corresponding lines in *message* (*iret* = 4) container:

  characters other than the alphabet A-Z and a-z, figures 0-9 and the signs: ,) (* + - =/ '& $! %; é è with _

- For histor :

  Detection of the lines of more than 72 characters and inscription of the corresponding lines in *message* (*iret* = 4).

  Detection of the lines containing a character other than: any character authorized in the various types of source except the accentuated characters and inscription of the corresponding lines in *message* (*iret* = 4).

- Scan for the noun of the units of source according to the criteria given for the task B ofasno.
- Research in the file of code noted of the already noted units of source.
  - If to use is not the only one to have noted the unit of source:

    if it is the first a message indicating the name of the unit is registered in *message* (*iret* = 0),

    if not a message indicating the name of the unit and the name of the noted author first is registered in *message* (*iret* = 4),

    If a unit were not noted a message indicating the name of the unit is registered in *message* (*iret* = 0) and (very nicely) it is noted.
- Checking of coherence with the version of reference *version* :
  - Checking that units of source of FORTRAN or cal or c/*.c or catalogu :

    type MODIFICATION are restored in the good library and at the same date as the version of reference *version*. ,

    type ADDITION already do not exist in all the libraries of*Aster* and are restored in an existing library.
  - Checking that specified libraries of the units of source of unigest to destroy are correct for the version *version*.
  - Checking that specified libraries of the units of source of unigest with to move are correct for the version *version*.
  - Checking that CAS-tests:

    type MODIFICATION exist and at the same date as the version of reference *version*,
    type ADDITION already do not exist

If such is not the case, the accused units of source are announced in the file `message` (`iret = 4`).

- Destruction of all the repertoires and all the files being in the repertoire "`$TMPDIR/to use (S)`".

9.5 asrest : restitution of source

To check that a set of source can be introduced into a version of reference of *Code Aster* and to prepare the possible update.

This tool carries out a certain number of checks then it launches the passage of the CAS-tests of reference in a job (of modifiable night by default but).

This tool must be used by any person wishing to restore a development.

Like asverif, asrest allows to carry out a restitution grouped for the dependent developments.

9.5.1 Mode of call

asrest *repertoire* [*version* [*message* [classe_job]]]

repertoire is the name of a repertoire. It has the same function as in asverif. One can thus find there, according to the cases, in repertoires Use-CRAY files:

- histor,
- FORTRAN,
- cal,
- catalogu,
- unigest,
- test
  - *.comm Command file of the CAS-test,
  - *.mail Grid Aster of the CAS-test,
  - *.mali Grid Ali-Baba of the CAS-test,
  - *.mgib Gibi grid of the CAS-test,
  - *.msup Grid Super-CAS-test,
  - will *.para Parameters of execution of the CAS-test.
- c/*.c Files of the functions C.

Compared to asverif one must find in the repertoire test all files new or modified concerning the added or modified CAS-tests.

version : is a parameter which is called after the version of reference concerned (NEW1, NEW2, NEW3). The value taken by default is NEW3.

message : definition identical to asno. The value taken by default is message_asrest.

classe_job : class of jobs CRAY subjected by as.tout.

9.5.2 Tasks carried out

Only (S) it (S) file (S) histor is (are) obligatory (S), the actions described below, of course, are carried out that if the files to which they refer exist. In certain case, actions of substitution will be carried out, they will be clearly specified.

- Tasks of asverif except the destruction of the repertoires of work
- If *iret* ≥ 4 END.
- Recording of the complete repertoire of *repertoire/to use (S)* in (S) the file (S) eda/*version/to use (S)*/identifi
- Insertion in to use (S) /histor of the author and current date.
- Update of the dates and the authors in the lines INFORMATION :
  - lines ADDITION are changed into MODIFICATION,
  - the dates are put at the date of use ofasrest,
  - the author carried by the line is put at *to use* more the name of the developer.

  Each update is the object of an inscription in the file of recapitulation eda/version/to use/recap
- Bursting of the routines of *to use (S)*/fortran in the repertoire eda/*version*/*to use (S)*/bibfor/*bibli* with a routine by file, *bibli* being the library specified by the line INFORMATION.
  Recopy of the routines to be moved FORDEPLA old library of reference in eda/*version*/*to use*/bibfor/*bibli* where *bibli* is the new library.
- Compilation of the routines of eda/*version*/*to use (S)*/bibfor, with: cft77_aster dvpdbg
  ERRORS of compilation are recopied in message (*iret* = 4), them WARNINGS also (*iret* = 2).
  The module object is preserved in fortran.o .
  (In the case of dependent developments there is a copy in each eda/*version*/*to use* concerned.)
- If *iret* ≥ 4 END.
- segldr with the libraries objects of the version version. The achievable module is called execut.
  (In the case of dependent developments there is a copy of execut in each eda/*version*/*to use* concerned).
- Compilation of the catalogues (tasks defined for ccat92 less the checks, - which are already carried out in asrest).
- If *iret* ≥ 4 END.
- To recopy in eda/*version*/*to use (S)* the list of the CAS-tests has to pass (liste_ct.rest) and to enrich it by the CAS-tests present in this evolution.
- Launching of the CAS-tests (by default in class harms).
  Tasks identical to as.tout with the files create by asrest (execut, catalogues orders and elements if they are present).
  The class of the jobs of CAS-test by default the batch of night (if not parameter *classe_job*).
  The launching of the CAS-tests is stopped so at least 5 CAS-tests do not end in OK.
  To pass the tests a repertoire eda/*version*/*user_de_reference*/astout is created. In this repertoire are all the files of CAS-tests necessary. (Those which overload the reference). Results of the CAS-tests not OK will be also in this repertoire under resu_NOOK/ and resu_NOOK/flash. (cf as.tout).

user_de_reference is it to use the appealing one if it returns him even of the source, if not it is the first to use alphabetically developers which restore.
Files .code and .resu CAS-test added are preserved. The information for the update of the list of the CAS-tests (total time, necessary memory) of the added or modified CAS-tests are stored.

- Recopy of the file *message* in (S) the file (S) eda/*version*/*to use (S)*/message.
  Recopy of the value of *iret* in (S) the file (S) eda/*version*/*to use (S)*/iret.

9.5.3 Files created

The files created are:

```
/      /eda/      / to     /histor
aster  version    use
                  (S)
                           /
                           fortran
                           /cal
                           /
                           catalog
                           u
                           /test    / nom_cas_test
                                    .comm
                                    / nom_cas_test
                                    will.para
                                    / nom_cas_test
                                    .mail
                                    / nom_cas_test
                                    .mali
                                    / nom_cas_test
                                    .msup
                                    / nom_cas_test
                                    .mgib
                           /
                           unigest
                           /bibfor  / bibli          / routine.f
                           /bibcal  / bibli          / module.cal
                           /bibc    / bibli          /
                                                     fonction_C.c
                           /obj     /fortran.o
                                    /cal.o
                                    / fonction_C .o
                           /execut
                           /catalo  / bibli          /
                                                     subcat.cata
                           /catobj  /commande
                                    /elements
                           /
                           identif
                           i
                           /
                           message
                           /iret
                           /recap
                           /ls_ct
```

```
/astout  /*.comm
         will/*.para
         /*.mail
         /*.mali
         /*.msup
         /*.mgib
         /liste_ct
         /resu_NOOK      /*.resu
                         /*.code
                         /flash/
                         *error

                         *output
```

9.6 as.tout : passage of CAS-tests

Sequence of a list of CAS-tests with analysis of the results.

Principal characteristics of this tool:

- construction of scripts of Unicos orders of each CAS-test starting from the data files *Aster*,
- tender of these CAS-tests by using the possibility of the CRAY of carrying out several jobs at the same time,
- collection of the results of the various jobs,
- realization of a synthetic document of the whole of the CAS-tests sent and possibly execution of an action according to the result of the CAS-tests.

Example of synthesis:

```
                          -- WITH S T E R -- VERSION  2.05.11 --

   --- End of all the cases tests       :  the 7/12/93 has
15:21: 01
   --- Here results:

                                      Time CPU  Time SYS  Total
time   Cost
   adlv100a    OK                       5.28       0.92       6.20
13.79 F
   ahlv100b        NOOK_TEST_RESU       6.70       0.65       7.35
16.33 F
   hplv100a    OK                       7.80       0.63       8.42
18.72 F
   hplv101a    OK                       4.34       0.72       5.05
11.23 F
   hplv101b        CPU_TIME_LIMIT       6.29       0.82       7.11
15.80 F
   hsnl100a    OK                       3.04       0.67       3.71
8.24 F
   hsnv100a    OK                      10.81       0.69      11.50
25.56 F
   --------  ----  ---------------  --------  --------
--------  --------
      7        5          2          44.26       5.09      49.35
109.67 F


                        PROBLEM WITH 2 CAS-TESTS
```

as.tout check the coherence of the transmitted arguments then built and subjects one job (qastout) same class as that of the CAS-tests. It job built and subjects chains of CAS-tests and expects the end of all the CAS-tests launched to format the results and to send them in the file message.

9.6.1 Mode of call

as.tout *fichier_parametre* [*message*]

fichier_parametre : File containing the different ones parameters of execution ofas.tout.

message : file of the messages. By default message_astout.

fichier_parametre contains 12 values.

A parameter is the first field of a line not begin with the character '%'. There cannot thus be several parameters on the same line.

The recognition of the parameters is done on the order of declaration. If one wants to have the value by default of a parameter it should be replaced by the character `'.'`. The blank lines are authorized.

The parameters are the following:

- name of the file containing the list of the CAS-tests (not value by default),
- repertoire containing the files reference of the CAS-tests (command files, grid and parameter with the same organization that in the reference index by default `/aster/astest/`*`version`*`/-`),
- repertoire containing of the files of CAS-tests which will overload the files of reference (not value by default),
- repertoire receiving the results (not value by default), the files `nom_cas - test.resu` and `nom_cas-test.code` there are create and, if there already does not exist, a repertoire `flash` is created, it receives the files `nom_cas-test.error` and `nom_cas - test.output`),
- name of the achievable one *Aster* (by default `/aster/`*`version`*`/execut`),
- file containing the catalogue of the compiled orders (by default `/aster/`*`version`*`/catobj/commande`),
- file containing the catalogue of the compiled elements (by default `/aster/`*`version`*`/catobj/elements`),
- the class of job CRAY to pass the CAS-tests: `batch` or `nbatch` or `wbatch` (by default `nbatch` - batch of night),
- the maximum number of CAS-tests which finish badly before interruption of the launching of the other CAS-tests (not of defect),
- the version *Aster* for the CAS-tests (is used to determine the names by default, `NEW3` by default),
- if "`DATE`" the date and the hour are added in extension of the results of the jobs, if not the extensions `.error` and `.output` are added in the name of the CAS-tests for the result of jobs CRAY,
- if "`RESOK`" the files of result of all the launched jobs are sent in the repertoire result; for any other value, only results of the jobs not `OK` are preserved.

9.6.2 Tasks carried out

By `as.tout` :

- Checking of the parameters:
  - Existence of the various repertoires and files.
  - Checking of the fields of validity of the parameters classifies job, version and dating of the files of the "flashor".
  - Checking which the names of CAS-test respect the rule of naming (letters then figures and a letter).
  - If `iret` ≥ 4 END.
- Checking of the existence of the files `will.para` CAS-test to be passed.
- Construction of the job launcher of the same CAS-test classifies than the CAS-tests.
- Tender of this job by `qastout`.

By the launcher of CAS-test `qastout`:

- Checks of the first two tasks of `as.tout` .

- Creation of a repertoire in `/tmp` to collect the results of all the CAS-tests.
- Construction of the jobs of all the CAS-tests.

  Each job checks the following principles:

  - if parameters `NQS` are incompatible with the machine end of `as.tout` (`iret=4`),
  - if the CAS-test is not the last of a chain:
    - test the code return of `qsub` following CAS-test,
    - to check by `qstat` that the CAS-test is well taken into account by `NQS`,
    - to send one `e-mail` on CRAY with the administrator *Aster* (`d6bhhhh`) in the event of problem detected by `NQS`,
    - in the event of problem `NQS` call to script `Dernier_Job`.
  - if the CAS-test is the last of a chain:
    - call to script `Dernier_Job`.
- Tender of the CAS-tests in the shape of 16 sequential chains launched in parallel.

Script `Dernier_Job` is called by any CAS-test detecting a problem `NQS` and the last job of each chain. This script must determine if the job which calls it is the last CAS-test in execution on CRAY among all the CAS-tests launched by this `as.tout`. If it is the case `Dernier_Job` call upon the job `Dernier_Job.btc`

Tasks of the job `Dernier_Job.btc` :

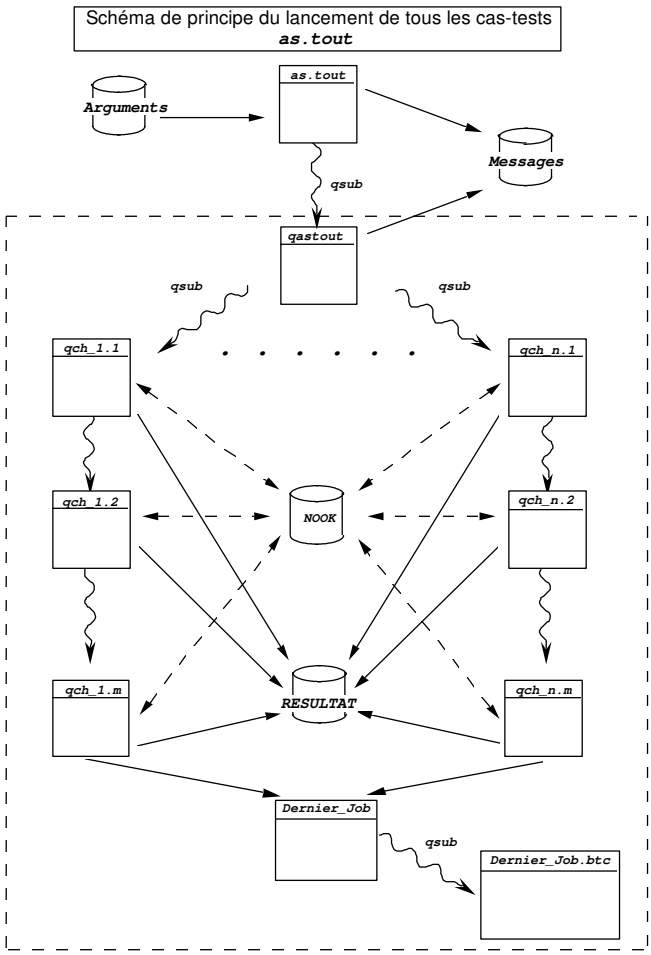
Working of the result of the CAS-tests

For each subjected CAS-test posting

- of a diagnosis:
  - OK,
  - `ARRET_ANORMAL` (stop not managed by *Aster*),
  - `<F>_SUPERVISEUR` (*Aster* stopped by the supervisor),
  - `ERREUR_<F>` (other fatal errors detected by *Aster*),
  - `ERREUR_<S>` crushing?????? without visible consequence,
  - `<E>_VOLATILE` there remains an object in the volatile base,
  - `NOOK_TEST_RESU` (not respect of the value of reference),
  - `DEFAUT_FICHIER` (it misses a file or a repertoire essential to the CAS-test),
  - `CPU_LIMITE` (lack of time to finish the CAS-test),
  - INTERRUPTION_JOB,
- time CPU,
- time system,
- total time of the CAS-test,
- cost of the CAS-test.

For the whole of the subjected CAS-tests posting:

- total time CPU,
- time total system,
- total time of all the CAS-tests,
- cost of all the CAS-tests.



9.7 ccat92 : compilation of the catalogues

Compilation of units of catalogue in command files and elements compiled.

Pas de checking of coherence, nor of identity of the developer.

9.7.1 Mode of call

ccat92 *catalogu catalo catobj* [*execut* [*para* [*version* [*message*]]]]

catalogu : file containing the units of catalogue to be compiled.

catalo : is a repertoire at exit which is organized and which contains the sources burst of catalogu in elements of CATALOGUE conform to the repertoire of the versions in exploitation.

catobj : is a repertoire at release in conformity with the repertoire of the versions in exploitation. catobj contains the catalogues compiled of the version *version* of*Aster* "overloadedES" by the elements contained in catalogu.

execut : file as starter which contains the achievable module which must carry out the compilation of catalogu (by default it is /aster/*version*/execut).

para : parameter which is worth "deprive" for the made use of ccat92 in asrest and which makes it possible to mark the catalogues compiled of the mention DEPRIVE.

version and *message* : definitions identical to asno (by default *message* = message_ccat92).

9.7.2 Tasks carried out

Checking that *catalogu* is well of the type CATALOGU (first task ofasno).

Bursting of the subcatalogues in catalo/
If *iret* ≥ 4 END

In repertoire work to establish a link symbolic system with all the catalogues of the version of reference.

To carry out the "overload" with the units of the file *catalogu* by recopying the files of *catalo*.

- Compilation of the catalogue of the orders:
  - Concatenation of all the subcatalogues of order in fort.3,
  - Assignment of the file 'COMMANDE_PRIVEE' with fort.3,

- Compilation of the file 'COMMANDE_PRIVEE' with *Aster* (*execut* or the achievable by default of *version*),
- If iret ≥ 4 END,
  Copy of the messages of compilation by *Aster* in message (fort.6)
  Copy of the file of error (fort.9) in *message* .
- If not result of the compilation of the orders saves in catobj/order.

.

- Compilation of the catalogue of the elements:
  - to recopy compelem/grandeurs_1ere.cata in fort.4,
  - to count the number of subcatalogues in options (=*nb_opt*),
  - to add 'OPTION_SIMPLE *nb_op*' in fort.4,
  - to add all the subcatalogues of options in fort.4,
  - to add 'END' in fort.4,
  - to add all the subcatalogues of typelem/ in fort.4,
  - to add 'END' in fort.4,
  - to add compelem/type_maille.cata in fort.4,
  - to add compelem/modes_partages.cata share in fort.4,
  - to add compelem/phenomenes_modelisation.cata in fort.4.

If *version* is NEW1 or STA1 there is no compilation with *Aster* . The result is right the concatenation described above.

If not compilation with *Aster* by using the catalogue of order previously compiled, if there exists.

If iret < 4 safeguard of the result of the compilation of the elements in catobj/elements.

If not:

- Copy of the messages of complation by *Aster* in *message* (fort.6).
- Copy of the file of error (fort.9) in *message*

10 Tools of update of the version

These tools will be available only on the machine of the administrator of the sources *of Aster*.

10.1 majnew : update of the version NEW

To update a version of reference NEW *Code Aster* by incrementing of level.

10.1.1 Mode of call

This module protected in reading, writing and execution is accessible only by the administrators from the versions.

majnew *l_developpor para version message iret_max user_IBM password_IBM*

l_developpor is a file in which one finds, line by line, the list of identifiers restoring their source.

para is a parameter which is worth "COPOLD" or "PASCOPOLD" allowing to save or simply to replace the up to date put version.

version is the version (NEW1 NEW2 NEW3) to update.

message **:** definition identical to asno. By default message_majnew.

iret_max **:** maximum error code of asrest so that the restitution is taken into account (0 or 2).

user_IBM: to use IBM for the safeguard.

password_IBM : password of using IBM.

10.1.2 Tasks carried out

If *version* = NEW2, is validity of the password on IBM. If problem *iret* = 4 and end of majnew

Constitution in /aster/eda/*version*/majnew/ files:

```
/aster/ eda/ version /majnew/ FORTRAN
                             histor
                             recap
                             liste_ct
                             ls_ct
                             catalo/ catastrophes/ under-
catalogue.cata
                             unigest
                             bibfor/ bibli/  routine.f
                             obj/    FORTRAN_user (S).o
                                     cal_user (S).o
                                     fonctions_C.o
                             test  *.comm
                                   will *.para
                                   *.mail
                                   *.mali
                                   *.msup
                                   *.mgib
                                   *.code
                                   *.resu
```

by concatenation of the corresponding files in the repertoires eda/*version*/*to use* of all the identifiers present in *l_developpor*.

This provided that the file eda/*version*/*to use*/iret exist and contains with more the value *iret_max*. If not a message recopying it eda/*version*/*to use*/message is emitted (*iret* = 4) and end of majnew.

If all them eda/*version*/*to use*/iret *l_developpor* are incorrect then END (*iret* = 4).

Sources FORTRAN compiled (in debug mode) of each *to use* are copied under the name fortran_*to use*.o, the assembler CAL compiled is copied under cal_*to use*.o and functions C under *nom_fonction*.o.

Incrementing of the number of level found in:

/aster/*version*/versio.

This file will be updated just at the end of the module.

Recopy with update of the date of the line INFORMATION and of the variables containing the number of version of the routine FORTRAN versio in bibfor.

Compilation in majdbg the module object is placed in:

/aster/ eda/ /*version* /majnew/ versio.o

Safeguard of the compiled libraries bibobj in bibold and catobj in catold if *param* = COPOLD.

Compilation of FORTRAN in majobj:

- Call to bld with the result for the update of the library bibobj with the suppressions of routines of unigest.
- Call to bld with fortran_*users (S)*.o for the update of the library bibdbg with the suppressions of routines of unigest.

Compilation of CAL in majobj :

- Call to bld with the result for the update of the library bibobj.
- Call to bld with cal_*users(S)*.o for the update of the library bibdbg.

Compilation of BIBC in majobj :

- Call to bld with the result for the update of the library bibobj.
- Call to bld with cal_*users(S)*.o for the update of the library bibdbg.

segldr with the library bibobj version *version*.

Update of /aster/*version*/execut

Destruction of the units of CATSUPPR in /aster

/aster/ *version*/ catalo/ *bibli*/ *sub_cat*
diffcat/*bibli*/ *sub_cat*

Compilation of the catalogues with the parameter *para* = "public" :

ccat92 *version* execut catalogu catalo catobj message_ccat iret_ccat *para*

diff-b/aster*version*/bibfor/*bibli*/*routine* /aster/eda/*version*/bibfor/*bibli*/*routine* for all the routines of majnew/bibfor

Concatenation, at the head, in /aster/*version*/diffsub/*bibli*/*routine* (with writing of an identification of the update).

diff-b/aster*version*/catalo/*bibli*/*sub_cat* /aster/eda/*version*/catalo/*bibli*/*sub_cat* for all the routines in majnew/catalo

Concatenation, at the head, in /aster/version/diffcat/bibli/sub_cat (with writing of an identification of the update).

Copy of

```
/      eda/  version      catalo/      bibli /  sub_ca
aster       /majnew/                           t
/
                          bibfor/      bibli /  routin
                                                e
                          bibcal/      bibli /  module
                          bibc/        bibli /  functi
                                                on
```

in :

```
/      version /       catalo/       bibli /  sub_ca
aster                                         t
/
                       bibfor/       bibli /  routin
                                               e
                       bibcal/       bibli /  module
                       bibc/         bibli /  functi
                                               on
```

Destruction of the routines to be destroyed unigest and of the old libraries of routines to be moved in :

/aster	/ version	/bibfor	/ bibli	/ routine
		/diffsub	/ bibli	/ routine
		/diffcal	bibli /	module
		/diffc	bibli /	function
		/fort	/ routine	

Creations of the links symbolic systems enters :

/aster	/ version	/bibfor	/ bibli	/ routine

and

/aster	/ version	/fort	/ routine

Copy of histor in

/aster	/ version	/histor	/v.sv.n

where v.sv.n is the number of version/under-version/level with heading: v.sv.nv version *date*

Addition of recap sorted alphabetically in the end of histor.

Determination and addition of the statistics in the end of histor.

Preparation ofhistor IBM version (by concatenation at the end of histor existing), for a version NEW2 only.

Suppression of all the notations of the units of FORTRAN, catalo, test and unigest in quirou, quicat and quites whatever the developers.

Suppression of all the notations of more than 3 months (including 'RAYE').

Destruction of the units TESSUPPR, and put up to date of the lists of CAS-tests (liste_ct.tout and liste_ct.rest).

Copy added or modified CAS-tests and put up to date of the list of CAS-tests to manage the lists, restricted, total and of performance: /aster/astest/*version*/Liste. The old list is preserved in the file /aster/astest/*version*/Liste.old

Rebootstrapping of the file of delegation of restitution lien_dvp

Update of /aster/*version*/versio

If*version* NEW2:

- copy on IBM added or modified files.
  Safeguard of the orders lay out of these copies in the file dispIBM (to be able to start again this operation independently moreover update in the event of problem withIBM).
- suppression on IBM files removed on CRAY (FORTRAN , CATALOGU and CASTEST).
  Safeguard of job IBM of suppression in the file suppIBM (to be able to start again this operation independently moreover update in the event of problem withIBM).

To send the file of messages of majnew with ASA by e-mail (Administrator of the sources *Aster*).

To send the file histor with all the developers having an account on a workstation (of which the address is in the file ident *Aster*).

10.1.3 Files of /aster/NEWn

```
/       version /         histor/         v.sv.n
aster
/
                          extremely       routine
                          bibfor/         bibli /   routine.f
                          diffsub/        bibli /   routine.f
                          bibcal/         bibli /   module.ca
                                                    l
                          diffcal/        bibli /   module.ca
                                                    l
                          bibc/           bibli /   fonction.
                                                    c
                          diffc/          bibli /   fonction.
                                                    c
                          diffcat/        bibli /   subcat.ca
                                                    ta
                          catobj/         order
                                          element
                                          s
                          versio
```

```
/       astest/ version  List
aster   /                Liste.old
/                        liste_ct.rest  castest
                         liste_ct.tout
                         diffct/
                         castest .comm
                         castest .mail
                         castest .mali
                         castest .mgib
                         castest .msup
                         castest
                         will.para
                         castest .resu
                         castest .code
```

10.2 finmaj : end of update of the version NEW

To destroy the files of the developers having restored their source (even those not having summers appointed for the update).

To save on IBM the files of the jobs of the update if the version = NEW2 .

This module protected in reading, writing and execution is accessible only by the administrators from the versions.

10.2.1 Mode of call

```
finmaj version message user_IBM password_IBM repertoire
```

version is the version (NEW1 NEW2 NEW3) who has just been updated.

message: definition identical to asno.

user_IBM IBM for the safeguard is to use it.

password_IBM is the password of using preceding.

repertoire is the repertoire which contains the files of result of majnew who must be saved on IBM.

10.2.2 Tasks carried out

- Destruction of /aster/eda/*version*/*to use (S)*
- Concatenation of the files present in *repertoire*.
- If *version* = NEW2 : safeguard on IBM of this file in ASTER.CRB (CVSVNV)
- If the safeguard occurred well, destruction of all the files of *repertoire*.
- Tender of the complete listing of the CAS-tests, night and by preserving only the result of the jobs Not OK (under /c/gr1/d6bhhhh/finmaj).

10.3 actulist : actualization of the cost of the various lists of CAS-tests

10.3.1 Mode of call

actulist [*sauv* [*version* [*message*]]]

sauv: if this parameter = "sauv" the result of actualization replaces the file Lists.

version : version of*Aster* concerned (NEW1 or NEW2 or NEW3 by default NEW3).

message: file of message of the order (by default message_actulist).

10.3.2 Tasks carried out

To be able to recover the rates of the various classes of job on CRAY actulist "is coated" in a job which:

- determine the rates of the jobs of day, night, weekend,
- Calculation of total times and the costs of the various lists of CAS-tests (restricted, supplements, performance),
- give the result in the file of message,
- if sauv= " sauv" :
  copy of the file /aster/astest/*version*/Liste
  in /aster/astest/*version*/Liste.old.

  Safeguard of the new file List :

  copy of the file /aster/astest/*version*/liste_ct.rest
  in
  /aster/astest/*version*/liste_ct.rest.old,
  copy file /aster/astest/*version*/liste_ct.tout
  in
  /aster/astest/*version*/liste_ct.tout.old,

  construction of the file /aster/astest/*version*/liste_ct.rest by extracting the CAS-tests having the indicator "." in last column of the file list.
- construction of the file /aster/astest/*version*/liste_ct.tout by extracting the cases - tests having the indicator "." or "S" in second column of the file lists. The cases - tests having the indicator "P" are in the file with one "%" in first column (to put them in comment).

10.4 majlist : update of the complete listings and restricted

10.4.1 Mode of call

```
majlist [version [message]]
```

version : version of*Aster* concerned (NEW1 or NEW2 or NEW3 by default NEW3).

message : file of message of the order (by default message_majlist).

10.4.2 Tasks carried out

- Update of the short list /aster/astest/*version*/liste_ct.rest while extracting from the file /aster/astest/*version*/Liste the fields name of CAS-test and title, the lines of which the second column (the separator being character |) a point contains.
- Update of the complete listing /aster/astest/*version*/liste_ct.tout while extracting from the file /aster/astest/*version*/Liste the fields name of CAS-test and title of all the lines describing a CAS-test. CAS-tests of the list of performance (one P in second column) are put in comment (one % in first character of the line).

10.5 majsta : stabilization of a version

Before the stablilisation of a version, to warn by mail all the people concerned of the date of the operation.

Stabliser the version allows NEW*N* in STA*N*.

10.5.1 Mode of call

majsta *numero_version* message para

numero_version : number of the version of*Aster* with stabliser.

message: file of message of the order.

para: parameter OLD to copy STA in OLD and PASOLD not to copy it.

10.5.2 Tasks carried out

Incrementing of 1 of the number of under-version in the routine versio

Compilation and edition of the links:

- maj of /aster/NEW*N*/bibobj
- maj of /aster/NEW*N*/bibdbg
- maj of /aster/NEW*N*/execut

If para = OLD Copy of /aster/STA*N* in /aster/OLD*N*:

- link between /aster/OLD*N*/execut and /aster/OLD*N*/EXECUT
- link between /aster/OLD*N*/catobj/elements and /aster/OLD*N*/CATOBJ/ELEMENTS
- link between /aster/OLD*N*/catobj/commande and /aster/OLD*N*/CATOBJ/COMMANDE

Copy of:

- /aster/NEW*N*/mode_exec

- /aster/NEW*N*/execut in /aster/STA*N*/execut
- /aster/NEW*N*/catobj/ * in /aster/STA*N*/catobj
  link enters `/aster/STAN/execut` and `/aster/STAN/EXECUT`
  link enters `/aster/STAN/catobj/elements` and `/aster/STAN/CATOBJ/ELEMENTS`
  link enters `/aster/STAN/catobj/commande` and `/aster/STAN/CATOBJ/COMMANDE`
- Conservation of the CAS-tests and their results
  `/aster/astest/NEWN/*.*` in `/aster/astest/STAN`

Creation of a histor of stabilization.

Update of /aster/NEW*N*/versio.

Send, by e-mail, of the histor to all the developers.

Safeguard on Sun cassette of:

/aster/NEW*N*/bibobj/
/aster/NEW*N*/bibdbg/
/aster/NEW*N*/fort/
/aster/NEW*N*/c/
/aster/NEW*N*/cal/
/aster/NEWn/catalo/
/aster/astest/NEW*N*/

10.6 Change of version NEW

Specific Pas d' tools for the change of version NEW.

Detail of the "manual" operations to carry out:

- To carry out a version STA with majsta.
- To save on cassette the version STA in particular files bibobj and bibdbg.
- Copy of the sources of NEW*N* has NEW*n+1* :
  to mkdir /aster/NEW *n+1*
  CCP - R /aster/NEW *N* /* /aster/NEW *n+1*
  to mkdir /aster/astest/NEW *n+1*
  CCP - R /aster/astest/NEW *N* /* /aster/astest/NEW *n+1*
- Copy of the restitutions carried out in NEW*N* to transfer them in NEW*n+1* :
  to mkdir /aster/eda/NEW *n+1*
  CCP - R /aster/eda/NEW *N* /* /aster/eda/NEW *n+1*
- To change the value by default of the parameter *version* for all the tools of the AGLA (modification of the routine /aster/adm/tool/init_defaut.csh).
  To make the same thing for asterix.
- To transform the notations NEW*N* in NEW*n+1* in the files:
  /aster/agla/qui/quirou
  /aster/agla/qui/quicat
  /aster/agla/qui/quites
- To modify the file /aster/eda/NEW*n+1* to put: *n+1*.0.0.
- To do one asrest on the CAS-test zzzz* to be able to make an update which will officialize the new number of version. (Not to forget to make a histor of circumstance).
- To make an update with majnew.
- To restrict the access to /aster/NEW*N* and /aster/astest/NEW*N* with aclaut.

11 Compilation and edition of the links

11.1 cft77_aster : official module of compilation of*Aster* for FORTRAN

To carry out the various modes of compilation in force for *Aster*. This centralization makes it possible to modify the official compilation easily of*Aster*.

If code return = 2 there are "warnings" FORTRAN for *Aster* or of the "warnings" not tolerated by *Aster*.

If code return = 4 there are errors of compilation FORTRAN or bad arguments.

The error messages are sent in the standard file of exit (it is obviously possible to redirect in a file with)

11.1.1 Mode of call

```
cft77_aster   mode_compil   file   object
```

mode_compil : mode of compilation of*Aster* development (= dvpdbg, dvpobj, majdbg, majobj).

file : source file FORTRAN to be compiled.

object : file result object.

11.1.2 Tasks carried out

Options of compilation in mode dvpdbg :
```
/bin/cft77_4.0 -i64 -dp -m3 -M 408,118,881 -eiz
```

Options of compilation in mode dvpobj :
```
/bin/cft77_4.0 -i64 -dp -m3 -M 408,118,881 -ez
```

Options of compilation in mode majdbg :
```
/bin/cft77_4.0 -i64 -dp -m3 -M 408,118,881 -ez
```

Options of compilation in mode majobj :
```
/bin/cft77_4.0 -i64 -dp -m3 -M 408,118,881
```

Besides the errors of syntax detected by the compiler, a certain number of "warning" is detected and involves a code of return 2 or 4 according to gravity.

Lists of the "warnings" with their number given by cft77.

The code return that they generate and a description succinte.

Their type: W warning

```
          With  ANSI

342   A2 :  Boolean been worth are nonstandard. (Prohibited
except DEFVEM)
720   A2 :  Equivalence character/not character
726   A2 :  Equivalence table length 1 and common
753   A2 :  The   uses   of   edict   descriptor   "item"   is
nonstandard. (Prohibited           except JEIMPO)
890   A2 :  Use of the tiny ones
895   A2 :  Variable   expression   has   year   assumed-length
character and is           year actual argument
```

All the others warning involve a code return = 4.

11.2 `cc_aster` : official module of compilation of*Aster* for the language `C`

To carry out the various modes of compilation in force for *Aster*.

The file object is the same one as the source file with the extension `.o`

If code return = `2` there are "warnings" `C`.

If code return = `4` there are errors of compilation `C` or of bad arguments.

The error messages are sent in the standard file of exit (it is obviously possible to redirect in a file with).

11.2.1 Mode of call

```
cc_aster   mode_compil   file
```

mode_compil : mode of compilation of*Aster* development (= `dvpdbg`, `dvpobj`, `majdbg`, `majobj`).

file : source file `C` to compile.

11.2.2 Tasks carried out

Options of compilation in mode `dvpdbg` :
`DC - C` **`- Gn`**

Options of compilation in mode `dvpobj` :
`DC - C`

Options of compilation in mode `majdbg` :
`DC - C` **`- Gn`**

Options of compilation in mode `majobj` :
`DC - C`

11.3 as_aster : official module of compilation of*Aster* for assembler CRAY (CAL)

To carry out the various modes of compilation in force for *Aster*.

If code return = 2 there are "warnings" ace.

If code return = 4 there are errors of compilation ace or of bad arguments.

The error messages are sent in the standard file of exit (it is obviously possible to redirect in a file with)

11.3.1 Mode of call

```
as_aster   mode_compil   file   object
```

mode_compil : mode of compilation of*Aster* development (= dvpdbg, dvpobj, majdbg, majobj).

file : source file assembler to be compiled.

object : file result object.

11.3.2 Tasks carried out

Options of compilation in mode dvpdbg :
ace

Options of compilation in mode dvpobj :
ace

Options of compilation in mode majdbg :
ace

Options of compilation in majobj mode:
ace

11.4 segldr_aster: official module of edition of the links of*Aster*

Constitution of new achievable *Aster* starting from libraries and of éventelles libraries (or files objects) of overload.

If code return = 2 there are "warnings" segldr.

If code return = 4 there are errors of edition of the links or bad arguments.

The error messages are sent in the standard file of exit (it is obviously possible to redirect in a file with)

11.4.1 Mode of call

segldr_aster *bib_aster bib_fermeture nv_execut* [*file (S)_surcharge*]…

11.4.2 Tasks carried out

Call to segldr with the libraries in the following order (decreasing priority):

- bin=*fichier_surcharges* if they exist,
- bin=biberon_aster
- bin=biberon_fermeture
- bin=/usr/lib/sysz%nag12
- bin=/usr/lib/sysz%generale

Options of segldr :

- - F zero Initialization of the variables with zero.
- - D 'DUPENTRY=NOTE, NOTE, NOTE' Pas d' posting of the messages for the entries of modules in double (because of the library of overload fermetur)

12 Management tools of the AGLA by the administrator

12.1 `bolt` : locking of the AGLA for the developers

Is used to prohibit the access to the tools of the AGLA which modifies the files of reference. That makes it possible to suspend the use of the AGLA for the update of the AGLA.

12.1.1 Mode of call

```
bolt   ["explanatory message of the cause of locking"]
```

12.1.2 Tasks carried out

Met places from there the mechanism of locking of the AGLA. The explanatory message will be transmitted to any developer which will try to use one of the tools of the AGLA using the mechanism of standard locking.

12.2 `deverrou` : unlocking of the AGLA

Allows to unbolt the mechanism of exclusive access to the tools of the AGLA. Is useful after a "planting" of one of these tools or after a voluntary locking of the Administrator (with `bolt`).

Also allows each developer to remove a bolt relating to sound `to use`.

12.2.1 Mode of call

```
deverrou
```

12.2.2 Tasks carried out

Unlocking of the AGLA.

12.3 `fgrep_agla` : "fast-grep" in the sources of the AGLA

To search a simple character string (not a regular expression) in the sources in C-SHELL.

12.3.1 Mode of call

```
fgrep_agla 'character string'
```

12.3.2 Tasks carried out

```
fgrep  "character string" /aster/outils/ * [a-z0-9]
/aster/adm/tool/ * [a-z0-9]
```

12.4 `stat_agla.qsub` : trees of static call of scripts-shells

To give for each tool of the AGLA the tree of call of the C-SHELL modules and C.

Results in `/aster/adm/g_agla/resu`.

Bug : do not go for dynamic calls (by variables).

12.4.1 Mode of call

```
qsub stat_agla.qsub
```

This operation being relatively long it is preferable to send it in batch (with `qsub`).

12.4.2 Tasks carried out

For each module of the AGLA, to search the calls:

- `source` : of Shell in "include",
- `HSC` : of C-SHELL in autonomous module,
- `qsub` : of C-SHELL in batch,
- direct calls: programs written out of C.

12.5 `app_agla.qsub` : appealing of script-SHELL

To search for each module of the AGLA the modules which refer to him.

Results in `/aster/adm/g_agla/resu/liste_appelant`.

Bug : do not go for dynamic calls (by variables).

12.5.1 Mode of call

```
qsub app_agla.qsub
```

12.5.2 Tasks carried out

For each module of `/aster/adm/tool` research in `/aster/adm/tool` and `/aster/outils` modules which refer to them.

13 Tools for *Aster*

13.1 Safeguard of the sources on IBM: limited to `NEW2`

13.1.1 `dispose_fort_agla` : safeguard of source FORTRAN

Copy of the files contained in `/aster/NEW2/bibfor` on `ASTER.NEW2.FORT ()`.

Caution :

It is necessary to update the password of using IBM `GJMHHTS` .

It is preferable to send the order in batch.

13.1.2 `dispose_cat_agla` : safeguard of source CATALOGU

Copy of the files contained in `/aster/NEW2/catalo/`*`catalogue`* on:

`ASTER.NEW2.CATALO.`*`catalogue`* `().`

Caution :

It is necessary to update the password of using IBM `GJMHHTS` .

It is preferable to send the order in batch.

13.2 Safeguard of the sources on Sun cassette

13.3 aclaut Access controls of files and repertoires

Use of the features ofAccess Control List (ACL) environment Multilevel Security (MLS) system UNICOS 7.0.

WithACL it is possible to manage the access control to files and repertoires with other tools that the permissions read, Write, execut for to use, group or other.

While being owner of a file it is possible to invalidate or add one of these permissions to one to use or a set of to use, without having to modify their membership of one group (what is reserved to the super-user).

The order aclaut allows to apply to one or more files or repertoires this access control.

13.3.1 Mode of call

To position in the repertoire /aster/agla/acl
Cd /aster/agla/acl
aclaut *nom_fichier_aut*

The file aut is a file of type text where one gives:

- the list of the files or repertoires to which one must apply an access control,
- the list of the authorizations or prohibitions.

Lines white or beginning by % are regarded as comments.

The lines giving the files of application are form:

```
FIC>nom_de_fichier_ou_repertoire
```

Only one file name per line. If a name of repertoire is given, the access control will be applied recursively to all the files and repertoires which it contains.

The lines giving the authorizations or prohibition have form:

```
ACL>autorisation_au format_acl
```

Only one authorization per line.

Example of file `aut` to give access only to the users `gjbhhts` and `j2bhhmb` (apart from the owner `d6bhhhh`) with the source files of the version `NEW2` of*Aster*.

```
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
        % File of description of the access control to the
sources of NEW2
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%

        %---------------------------------------------------
        % Lists files of which the access should be controlled
        %---------------------------------------------------
        % One by line precedes by FIC>
        % If one gives a repertoire the access control is
bracket
        % recursively has all its entries.
        FIC>/aster/NEW2
        FIC>/aster/astest/NEW2

        %---------------------------------
        % Description of control by using
        %---------------------------------
        % an access control per line, in the form
        % ACL>a: to use: group: permissions_d_acces
        % permissions of access: R read, W Write, X execut, N
nun

        % To prohibit all users
        ACL>a: *:   : N

        % To authorize the following users:
        ACL>a: gjbhhts: *: X-ray:
        ACL>a: j2bhhmb: *: X-ray:
```

13.3.2 Tasks carried out

Checking of the existence of the file *`fichier_aut`*.

Construction of a file `/aster/agla/acl/`*`fichier_aut`*`.acl` by the recovery of the control statements `ACL` and the use of the order `spacl`.

To apply recursively to all the files and repertoires requested this file `aut` by the order `spset`.

To preserve the trace of these orders in the file `/aster/agla/acl/`*`fichier_aut`*`.log`.

Intentionally white left page.