

Instruction manual of the AGLA for the administrator

Summary:

This document describes the interfaces of the tools of the AGLA which is accessible only to the administrator from the sources *Aster* (ASA).

That relates to primarily the tools of update of a version NEW and those of stabilization of a version.

Contents

Contents

1	Tools of the administrator of the sources Aster (ASA) on the machine of development.....	3
1.1	Organization of the tools.....	3
1.2	asdeno_asa : interactive denotation by the administrator.....	3
1.3	as.tout_ref : passage of tests for a version of reference.....	3
1.4	prepa_majnew : update of version NEW.....	3
1.5	prepa_finmaj : end of update of version NEW.....	4
1.6	creation of the catalogues of orders and elements.....	4
1.6.1	ccat_com.csh : creation of the catalogue of compiled orders.....	4
1.6.2	ccat_ele.csh : creation of the catalogue of compiled elements.....	4
1.6.3	make_cata_ele.pickled.sh : creation of the catalogues of elements.....	4
1.7	actulist : actualization of the cost of the various lists of case - tests.....	5
1.8	majlist : update of the complete listings and restricted.....	5
1.9	majsta : stabilization of a version.....	5
1.10	sauv_aster_tar : safeguard of a version.....	5
1.11	bolt : locking of the AGLA for the developers.....	6
1.12	deverrou : unlocking of the AGLA.....	6
2	The update of a version NEW.....	6
2.1	Launching on the waiter of development by demon.....	6
2.2	The construction of the parallel versions.....	6
2.3	The construction of the peripheral versions.....	7
3	Lbe utility.....	7
3.1	Construction of the files associated with tool CRP.....	7
3.2	Construction of one version using the tools of the AGLA.....	7

1 Tools of the administrator of the sources Aster (ASA) on the machine of development

1.1 Organization of the tools

The tools of the agla are scripts. (C-SHELL, korn_shell, bash) or of achievable deposited under the tree structures `/aster/agla/bin` and `/aster/agla/tool`. The source files of achievable are deposited under `/aster/adm/agla` on the waiter of reference.

1.2 `asdeno_asa` : interactive denotation by the administrator

Possibility only given to the ASA to in full safety publish the files of notation of the sources. While the file is published, an UNIX bolt is posed on the file. This bolt sets up a queue for any reading or writing on the file of notation. It is raised as soon as the editor is closed.
The administrator intervenes directly in the file to remove the noted units.

```
asdeno_asa - fn fichier_notation [- E editor]
```

`fichier_notation` indicate the file of notation: `quirou` (FORTRAN, FORTRAN90, C), `quicat` (catalogues), `quites` (CAS-tests), `quipyt`(python).

`editor` is the order to launch the editor of his choice (by default `nedit`).

1.3 `as.tout_ref` : passage of tests for a version of reference

```
as.tout_ref fichier_parametre [message]
```

`fichier_parametre` : file containing the various parameters: identical to that of `as.tout` [D1.02.01].

`message` : file of messages. By default `message_astout`.

This tool, reserved for the ASA, makes it possible to pass a list of CAS-tests with the heading of an official version not overloaded in the files of messages, and updates the files of messages and results (`.code`, `.mess`, `.resu`) in the version of reference (`/aster/vx/version/astest/`).

Several files of parameters and scripts of launching in batch are available under the account of administration on the waiter of reference (`/home/aster/astoutref`).

1.4 `prepa_majnew` : update of version NEW

To update a version of reference `NEW Code_Aster` by incrementing of level.

```
prepa_majnew version to use l_developpor para dir_trav action
```

`version` is the version (`NEW10` with `NEW1x`) to update.

`to use` is the account of administration used.

`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.

dir_trav : file of messages. By default *message_majnew*.

action : is worth *edition* or *tender*. In the first case the file script is generated then published. In the second case the file script is generated then subjected.

1.5 *prepa_finmaj* : end of update of version NEW

The tool *prepa_finmaj* allows:

- to destroy the files of the developers having correctly restored their source, the repertoire of a restitution which failed is left in the state so that the developer can become acquainted with the files of messages of the tests which do not function.
- to launch the lists of CAS-tests with update of the files of messages and results in the reference (*as.tout_ref*),
- to carry out scripts to synchronize versions on other platforms of execution and launching the cases tests on these last.

```
prepa_finmaj rep_tool rep_eda rep_bin version
```

```
rep_tool  repertoire of scripts of the agla ( /aster/agla/tool ),  
rep_eda   repertoire of the various restitutions ( /aster/eda/ version ),  
rep_bin   repertoire of the principal procedures of the agla ( /aster/agla/bin ),  
version   is the version ( NEW10 with NEW1x ) who has just been updated.
```

1.6 creation of the catalogues of orders and elements

Various scripts are normally used in the environment of the update of a version of development, and simplest is to use a profile *astk* (*/aster/adm/g_agla/make_catalo.export*) to build the catalogues of orders compiled and JEVEUX of the catalogues of elements bases it, the main difficulty residing in the reorganization of the source files.

1.6.1 *ccat_com.csh* : creation of the catalogue of compiled orders

The tool *ccat_com.csh* allows to create the catalogue of orders.
This script is usable only in the environment of *prepa_majnew*.

1.6.2 *ccat_ele.csh* : creation of the catalogue of compiled elements

The tool *ccat_ele.csh* allows to create the catalogue d'éléments.
This script is usable only in the environment of *prepa_majnew*.

1.6.3 *make_cata_ele.pickled.sh* : creation of the catalogues of elements

The tool *make_cata_ele.pickled.sh* allows to build the file *cata_ele.pickled* who is used to build in an incremental way to build the file *catobj/elements*.

1.7 actulist : actualization of the cost of the various lists of case - tests

To recompute times and costs of the various lists.

```
actulist [sauv [version [message]]]
```

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

version : Code_versionAster concerned (NEW10 with NEW1x).

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

1.8 majlist : update of the complete listings and restricted

To update the files liste_ct.rest, liste_ct.tout, liste_ct.perf and liste_ct.vali, starting from the file List.

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

version : Code_versionAster concerned (NEW10 with NEW1x).

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

This utility is not used any more by the procedures of update, it is /aster/agla/tool/make_listes.csh who is used in prepara_majnew.

1.9 majsta : stabilization of a version

Allows to stabilize the version NEWN in STAN :

- copy of STA in OLD if asked,
- update of the n° of version of NEW,
- copy of all the achievable files, sources and CAS-tests of NEW with STA.

```
majsta numero_version message para
```

numero_version : number of the version of Aster to stabilize.

message : file of messages of the order.

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

This utility is called in /aster/agla/outils/majsta.btc to be launched in batch.

1.10 sauv_aster_tar : safeguard of a version

Script /aster/agla/outils/sauv_aster_tar allows to save a version in a file tar to build an image to be engraved on an external medium.

```
sauv_aster_MMN_tar version rep_public lib_med lib_hdf standard
```

version : version of Aster to save.

rep_public : repertoire containing the external bookstores public.

lib_med : name of bookstore MED,

lib_hdf : name of bookstore HDF,

type : source or test is worth according to the type of the files to be filed.

1.11 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. (Useful for the update of the AGLA, for example.)

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

1.12 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).

2 The update of a version NEW

2.1 Launching on the waiter of development by demon

The update is launched automatically on the waiter of development under the account of administration (aster) via a crontab:

```
aster@frontall: ~$ crontab - L
# m H dom my dow command

# launching of the update
# -----
#
#
# versions of development
# =====
#
# version of development
40.0,2 ** 3.4 /home/aster/majnew/prepa_majnew_dvp.csh > /dev/null
...
```

For the update, the administrator must draw up the list of the developers having stated to carry out a restitution and to deposit it under `/aster/eda/version/ldvp`. This file is built starting from the list of the accounts having presented a restitution at the time of the EDA (`/aster/pre_eda/version`). The procedure of update will check that all the restitutions correctly proceeded (`iret=0` or `iret=2`) and will subject then the script of update `prepa_majnew`.

Script `/home/aster/majnew/prepa_majnew_dvp.csh` prepare a file which can be subjected in batch on the waiter of reference.

2.2 The construction of the parallel versions

At the time of the update, if the construction of the sequential version succeeds, parallel version MPI is built starting from script `/aster/agla/tool/make_mpi.sh`:

```
/aster/agla/tool/make_mpi.sh version
```

```
version : Code_version Aster concerned (NEW10 with NEW1x).
```

2.3 The construction of the peripheral versions

At the time of the update, if the construction of the sequential version succeeds, the various peripheral versions are built starting from script `/aster/agla/tool/maj_versions_locales.sh`:

```
/aster/agla/tool/maj_versions_locales.sh version
```

```
version : version Code_Aster concerned ( NEW10 with NEW1x ).
```

3 Lbe utility

3.1 Construction of the files associated with tool CRP

The construction or the modification of the files associated with the tool with checking with the rules with programmation (CRP) can be realized using the utility deposited under the account administrator `aster : /home/aster/CRP/make_crp_lsf.btc`. This file is submitted to the manager of batch, it works with the sources of the version of reference `/aster/v11/NEW11` and files deposited under `/aster/v11/NEW11/crp`. In return of the execution, the repertoire `/home/aster/CRP/NEW11/base` contains files which must be moved manually under `/aster/v11/NEW11/crp/base`.

3.2 Construction of one version using the tools of the AGLA

It is possible to carry out a complete recompiling of the sources of a version with the tools of the AGLA, i.e. by using the options of compilation of the version of reference and the tools CRP (control of the rules of programming) and CRS (control of the rules of syntax).

```
/ aster/adm/g_agla/compil_tout.ksh - towards version - reference mark  
rep_trav - msg fic_mess
```

`version` is the version (`NEW10` with `NEW1x`) with recompiler,
`rep_trav` repertoire of work, the result (achievable and bookstores) there will be deposited,
`fic_mess` file of the messages of compilation and edition of the links.