

Manuel de Descriptif Informatique
Fascicule D2.01 : Surcharge d'une version en exploitation
Document D2.01.01

Règles concernant l'extraction et la restitution des sources d'Aster

Résumé :

On donne ici les règles que doivent respecter les développeurs d'Aster lorsqu'ils récupèrent un source officiel (extraction) et lorsqu'ils proposent une modification d'un source officiel (restitution). Ces règles précisent des points également abordés dans le plan de développement [A2.01].

1 Introduction

On a expliqué dans [D1.01.01] comment un développeur pouvait "surcharger" le *Code_Aster* avec des sources personnels. Les sources dont on parle ici sont :

- des routines FORTRAN, ou C ou CAL (CRAY Assembling Language),
- des catalogues,
- des cas-tests.

Les sources officiels du Code sont accessibles en lecture sur le CRAY [D1.02.01 §2].

Un développement classique d'*Aster* se traduit en général par :

- la modification de certains sources,
- l'écriture de nouveaux sources.

La modification de sources suppose leur extraction préalable (recopie) des sources officiels. Le paragraphe 2 concerne les **règles d'extraction**.

Le développement terminé, il peut être "restitué" : on met alors à jour les sources officiels. Le paragraphe 3 concerne les **règles de restitution**.

2 Règles d'extraction

- R1 L'extraction d'un source est la recopie de la version officielle [D1.02.01 §2] de ce source. (Eviter les copies de copies).
- R2 Lorsqu'un source doit être modifié en vue d'une restitution future, "noter" ce source [D1.02.01] (commande `asno` d'`asterix`). Cette notation donne des droits de préséance pour la restitution lorsqu'un source est modifié simultanément par plusieurs développeurs : le premier "noté" est celui qui a le droit de restituer.
- R3 Lorsqu'un source doit être modifié et restitué : **ne pas toucher** la "ligne" `MODIF` qui contient la date de la dernière modification du source. Cette date permet de vérifier lors de la restitution que le source d'origine (avant modification) est bien le **dernier** source officiel. Ceci évite qu'une modification "n'écrase" une modification précédente. Cette règle est **capitale** : une violation intentionnelle serait très mal interprétée ... et sanctionnée.
- R4 Ne pas modifier (pour le plaisir ou par goût) la présentation du source : l'indentation, le nom des variables, ... sauf si c'est pour le rendre conforme aux règles de présentation ou de programmation [D2.02] ; car ceci augmente artificiellement le volume des "delta d'évolution" que l'on conserve : les modifications pertinentes ne sautent plus aux yeux.
- R5 Si le source extrait a déjà été noté par un autre développeur :
- le contacter pour lui faire confirmer son intention de restituer ce source,
 - essayer de négocier avec lui, l'ordre des restitutions,
 - s'il reste noté "devant" vous, ne pas se lancer tête baissée dans une modification de grande ampleur de ce source, car il faudra recommencer les modifications dans le source officiel issu de son évolution (cf. R3).
- R6 Ne pas noter inutilement des sources : vous pouvez gêner les autres développeurs (cf. R5). En particulier, ne pas noter un source que l'on compte restituer dans 6 mois !
- R7 Lorsque l'on a noté un source par erreur, le dénoter [D1.02.01] (commande `asdeno`).

3 Règles de restitution

- R1 Utiliser le mécanisme automatique de restitution d'asterix/agla : commande `asrest` [D1.02.01].
Ce mécanisme impose qu'une restitution soit autonome : le code officiel modifié par la restitution doit être capable d'exécuter les cas-tests de non régression.
- R2 Un "asrest" en efface un autre : il n'y a pas de pile de restitution pour un développeur donné. Un développeur ne peut donc faire plus d'un `asrest` par semaine pour une version donnée du code (NEW2 ou NEW3).
- R3 Une "restitution" (les sources modifiés + les sources ajoutés) ne sera effective (modification des sources officiels) qu'après la procédure de mise à jour du code faite par l'administrateur [D1.02.02 §1.2].
- R4 Ne détruire les sources d'une restitution qu'après avoir reçu (et lu) le "mail" annonçant la mise à jour du code.
- R5 Toute restitution doit être accompagnée d'un fichier d'historique (type `histor` d'asterix) dans lequel le développeur doit documenter sobrement mais avec précision sa restitution :
- nouveaux mots clés (vocabulaire),
 - nouvelles fonctionnalités : un peu de mécanique et d'analyse numérique,
 - les numéros des anomalies corrigées : AL94-XX,
 - les numéros des évolutions traitées : EL95-XX,
 - les évolutions ou ajouts des cas tests.

Ne pas se fatiguer à lister les noms des unités de source modifiées : le système le fait automatiquement.

- R6 "présenter" sa restitution en réunion d'EDA le lundi. Apporter un transparent de son fichier historique.
- R7 Penser à la validation de sa restitution : ajout/modification de cas-tests.
- R8 Pour détruire un source (ou le changer de bibliothèque) voir [D1.02.01 §3].
- R9 Quand on ajoute une nouvelle unité de source. Cette unité doit contenir une ligne 'AJOUT' dont la syntaxe dépend du type de l'unité [D1.02.01 §1].

C AJOUT nom_de_bibliothèque pour du FORTRAN

%& AJOUT nom_de_catalogue (COMMANDE, TYPELEM, ...) pour du CATALOGUE

% AJOUT pour du cas test
% TITRE ...