

INTERFACE POM - MASCARET v3

Spécifications Fonctionnelles Détaillées

Réf. POM3-CS-2023-DS-PIM-015

	Nom	Société	Fonction	Date	Visa
Rédigé par :	M .Renon	CS Group	Équipe projet		
Validé par :	C. Mertz	CS Group	Chef de projet		
Pour application :	J. Covès	CS Group	Directeur de projet		

CS GROUP
6 rue Brindejonc des Moulinais
Parc de la Grande Plaine
BP 15872
31506 Toulouse Cedex 5

Ed.	Rév.	DATE	MOTIF
03	01	14/09/18	Création du document
03	02	24/09/18	Prise en compte des remarques
03	03	01/10/18	Prise en compte des remarques
03	04	04/10/18	Prise en compte des remarques
03	05	05/10/2018	Prise en compte des remarques
03	06	05/10/2018	Prise en compte des remarques
03	07	19/12/2019	Gestion des dates T0, T1, T2 ; Gestion des calculs d'Analyse et Prévision en fonction des modes mascaret ; Gestion des paramétrages ; Gestion des outputs
03	08	29/09/2020	Adaptations au PI 3.1
3.1	01	11/12/2020	[#198255] changement d'unité des « valeurMin » et « valeurMax » issues du fichier « ParametresAssim.xcas »
3.1	02	15/01/2021	[#198741] Calage des valeurs « m » et « n » en fonction des balises « valMinKS » et « valMaxKS »
3.2	00	14/10/2021	Analyse BC2 : intégration Mascaret V3 et utilisation d'une VM Mascaret sous Debian 10 ; suppression assimilation
3.2	01	02/11/2021	Analyse BC2 : initialisation RAMPE ; ajout d'un décalage vertical en m sur les hauteurs Prise en compte des remarques
3.2	02	29/11/2021	§1.6.2 : génération des fichiers de reprise, backuptimesteps §3.6.1 : initialisation RAMPE §3.6.5 : run d'analyse : choix du fichier ligne d'eau §3.7.3 : archivage de la ligne d'eau
3.2	03	02/03/2022	§1.6.2 : - clé 'mascaretmodelsdirectory' - 'backuptimesteps' : ajout du tri ASC et sans doublons - suppression des clés 'resultfilecasiers' et 'resultfileliaisons' - balise <LigEaulnit> au lieu de <repriseCalcul>
3.2	04	24/08/2022	§1.9.1 : fichier « xcas », balises « fichierCas » et « fichier_cas »
3.2	05	02/02/2023	Gestion des ouvrages et barrages Mise à jour des options des dossiers modèles (§1.6.2)
3.2	06	14/02/2023	Prise en compte des retours SCHAPI
4.0	00	18/08/2023	Suppression des anciennes clés ('modelname' et 'modelsubdirectory') (cf § 1.6.2, § 1.7.2) Mise à jour du compute mode (cf § 2.1)
4.0	01	02/04/2024	Issue #16 : reprise des casiers entre les runs §1.6.3.3 Fichier résultats des casiers « .casiers_opt »

			§1.6.3.4 Fichier archive des casiers « .casiers_pim » §3.6.4.3 Modification du fichier « ParametresMascaret.xcas » §3.6.4.4 Affichage dans le fichier « pom.log » §3.7.3 Archivage de la ligne d'eau
4.0	02	27/09/2024	Issue #19 reprise de l'état de seuil mobile §1.6.3.5 Fichier des cotes de crêtes « Fichier_Crete.csv » §1.6.3.6 Fichier archive des cotes de crêtes « .barrages_pim » §3.6.4.3 Modification du fichier « ParametresMascaret.xcas »

Sommaire

Table des matières

1. Généralités.....	6
1.1 Objet du document.....	6
1.2 Glossaire.....	6
1.3 Présentation globale du système.....	6
1.3.1 But.....	6
1.3.2 Contexte du système.....	6
1.4 Architecture générale.....	7
1.4.1 Architecture matérielle.....	7
1.4.2 Architecture logicielle.....	7
1.5 Version des composants.....	7
1.5.1 Version PI.....	7
1.5.2 Version POM.....	8
1.5.3 Version Mascaret.....	8
1.5.4 Version Python.....	8
1.6 Interfaces.....	8
1.6.1 Généralités.....	8
1.6.2 Paramétrage du PIM.....	8
1.6.3 Formats de fichiers.....	9
1.6.3.1 LOI.....	9
1.6.3.2 OPT.....	10
1.6.3.3 Fichier résultats des casiers « .casiers_opt ».....	11
1.6.3.4 Fichier archive des casiers « .casiers_pim ».....	12
1.6.3.5 Fichier des cotes de crêtes « Fichier_Crete.csv ».....	13
1.6.3.6 Fichier archive des cotes de crêtes « .barrages_pim ».....	13
1.7 Arborescence générale.....	13
1.7.1 Arborescences des fichiers Mascaret.....	14
1.7.1.1 Arborescence d'un modèle Mascaret.....	14
1.7.2 Arborescence de travail du PIM.....	15
1.7.3 Nomenclatures.....	16
1.8 Installation.....	16
1.9 Lancement de Mascaret.....	17
1.9.1 Modes de calculs de Mascaret.....	17
1.9.2 Régimes Mascaret.....	17
2. Adaptations du PI v3.....	19
2.1 Mode de calcul.....	19
2.2 Nettoyage des fichiers.....	19
2.3 Calculs en parallèle.....	20
2.4 Un « run » par scénario.....	20
2.5 Gestion de l'avancement.....	20
2.6 Gestion des unités.....	20
2.7 Dates des simulations analyse/prévision.....	21
2.8 Méthodes de démarrage de l'analyse à T0.....	22
2.8.1 Méthodes d'initialisation gérées par le PIM.....	22
2.8.2 Mode d'initialisation « PERMANENT ».....	22
2.8.3 Mode d'initialisation «REPRISE».....	22

2.8.4 Mode d'initialisation « DEFAUT ».....	23
2.8.4.1 Initialisation RAMPE.....	23
3. Périmètre fonctionnel du PIM.....	25
3.1 Lancement.....	25
3.2 Ordonnanceur de tâches.....	25
3.3 Initialisation.....	25
3.4 Traitement des scénarios.....	25
3.4.1 Run(s) d'analyse.....	26
3.4.2 Run(s) de prévision.....	26
3.5 Traitement des entrées.....	27
3.6 Gestion des Runs.....	27
3.6.1 LaunchProcessor : initialisation RAMPE.....	28
3.6.2 Types de run.....	28
3.6.3 Copie des fichiers du modèle.....	28
3.6.4 Paramétrage de la plateforme.....	28
3.6.4.1 Décalage vertical sur les données d'entrées.....	28
3.6.4.2 Génération des fichiers « .loi ».....	29
3.6.4.3 Modification du fichier « ParametresMascaret.xcas ».....	31
3.6.4.4 Affichage dans le fichier « pom.log ».....	32
3.6.5 Run d'analyse : choix du fichier ligne d'eau.....	33
3.6.6 Lancement du run.....	33
3.6.7 Lancement du run spécial barrages.....	34
3.6.8 Vérification de la bonne exécution du run.....	35
3.7 Traitement des sorties.....	35
3.7.1 Sorties de type « fichier ».....	35
3.7.2 Sorties XML Sandre.....	35
3.7.2.1 Décalage vertical sur les prévisions.....	36
3.7.3 Archivage de la ligne d'eau.....	36
3.8 Finalisation.....	38

Liste des tableaux

Tableau 1 : Glossaire.....	6
Tableau 2 : lancement des runs d'analyse.....	26
Tableau 3 : lancement des runs de prévision.....	27
Tableau 4 : archivages des lignes d'eau.....	37

Liste des figures

Figure 1 : Architecture matérielle.....	7
Figure 2 : dates des simulations.....	21

1. Généralités

1.1 Objet du document

Ce document présente la spécification logicielle du programme d'interface entre la plateforme opérationnelle pour la modélisation (POM) et le modèle Mascaret. Ce programme est nommé PIM (Programme d'Interface avec Mascaret).

1.2 Glossaire

Acronyme	Signification
BDH	Base de Données Hydro.
BDTR	Également appelée BDTR (Base de Données Temps Réel).
PHYC/PHYL	Également appelée PHYC/PHYL (Plateforme Hydro Centrale/Locale) pour inclure ses services web.
CS	Communication System
IHM	Interface Homme Machine
PI	Librairie PomInterface ou Programme d'Interface générique
PIM	Programme d'Interface avec Mascaret
POM	Plateforme Opérationnelle pour la Modélisation
SCHAPI	Service Central d'Hydrométéorologie et d'Appui à la Prévision des Inondations
SPC	Service de Prévision des Crues

Tableau 1 : Glossaire

1.3 Présentation globale du système

1.3.1 But

La POM est un outil du SCHAPI permettant de lancer des calculs de prévision des crues sur différentes plate-formes de modèles. Parmi ces plates-formes, certaines sont recommandées nationalement par le SCHAPI : Mascaret, Telemac, GRP, Plathynes.

La POM entre en interaction avec ces plates-formes selon un protocole standardisé. Les modèles doivent donc s'interfacer avec ce protocole pour interagir avec la POM.

Le présent document doit donc définir les méthodes et procédures à implémenter pour piloter Mascaret à partir des informations fournies par la POM, d'une manière semblable aux modèles ci-dessus.

1.3.2 Contexte du système

Conceptuellement, le serveur POM et le serveur « Mascaret » (machine virtuelle) sont des serveurs distants. Le présent logiciel est lancé à distance par la POM à l'aide du protocole SSH.

L'exécutable doit donc être lancé en ligne de commande avec pour paramètre

- ✓ le nom du fichier de paramétrage que la POM fournit (« parameters.xml »)
- ✓ le nom du fichier de paramétrage du PIM (cf. 1.6.2)

Le chapitre 1.4.1 détaille la chaîne d'exécutables nécessaires au lancement de Mascaret par la POM, via PIM (le présent exécutable). Du point de vue de la POM, cette distinction doit être transparente.

Cette spécification présente la version 3.2 du PIM.

1.4 Architecture générale

1.4.1 Architecture matérielle

Le programme d'interface doit être déployé sur un serveur Mascaret (machine virtuelle « Mascaret TR 2021 » de type Debian10), accessible en SSH depuis le serveur POM.

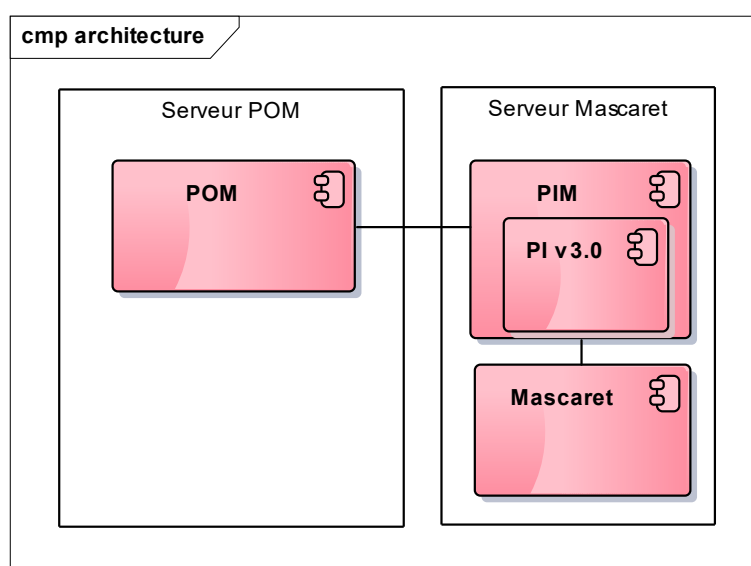


Figure 1 : Architecture matérielle

Le PIM repose sur la librairie PI (PomInterface) dans sa version 3.1 (au moins).

1.4.2 Architecture logicielle

L'architecture logicielle retenue est celle du Programme d'Interface générique PI également nommé librairie PomInterface. Le PIM consiste en une surcharge de certaines fonctions des processeurs.

1.5 Version des composants

1.5.1 Version PI

La présente spécification logicielle est prévue pour fonctionner sur la version 3.1 de PI.

1.5.2 Version POM

La présente spécification logicielle est prévue pour s'interfacer avec la version 2.3 du protocole POM.

1.5.3 Version Mascaret

La présente spécification logicielle est prévue pour s'interfacer avec Mascaret dans sa version v8p2r0.

1.5.4 Version Python

Les développements sont réalisés en Python 3.7.

1.6 Interfaces

1.6.1 Généralités

L'exécutable doit s'interfacer avec certains fichiers ou composants comme indiqué ci-dessous.

Le PIM est développé sur la base du Programme d'Interface générique (PI) développé pour réaliser les interfaces entre la POM et les modèles distants.

Le PI gère nativement l'interfaçage avec la POM :

- ✓ Arborescence de fichiers
- ✓ parameters.xml
- ✓ progression.xml
- ✓ ...

Le PI utilise la bibliothèque Libhydro pour la lecture / écriture des fichiers au format XML Sandre.

Note : le PIM n'exploite que des fichiers au format XML Sandre (en entrée et sortie). Les formats XML Image et JSON Lamedo ne sont pas pris en charge et une erreur le signale le cas échéant.

1.6.2 Paramétrage du PIM

Le fichier de paramétrage du **PIM** est au format « .ini » (pim.ini). Il reprend les paramètres du PI qu'il complète des paramètres ci-dessous, spécifiques à Mascaret, dans la section [pim].

Section des paramètres PI dont le contenu n'est pas celui par défaut :

- ✓ timeout : 300
- ✓
- ✓ rootdirectory : /home/mascaret

Les paramètres « commandline » et « commandlineworkdirectory » ne sont pas exploités par le PIM.

Cette section regroupe les paramètres du PIM.

✓ Paramètres « loi »

- ↳ stationsloi : liste des codes sites (8 caractères) et/ou stations (10 caractères) hydro qui fournissent des données pour les lois dynamiques de Mascaret (et pas les lois statiques), séparées par des « | ». Chaque station est un n-uplet dont les champs sont séparés par des « ; » : code site ou station ; nom de la loi correspondante dans le fichier ParametreMascaret.xcas ; décalage vertical en m.

✓ Génération des fichiers de reprise

- ↳ backuptimesteps (facultatif) : une liste de valeurs entières négatives, pour la génération de fichiers de reprise. Les valeurs sont séparées par des « | ». L'unité est la minute. Les valeurs doivent être triées de manière ascendante et sans doublons.

Si au moins une valeur n'est pas triée, une erreur est générée.

Si au moins une valeur est un doublon, une erreur est générée.

« 0 » est la valeur par défaut. Si « 0 » n'est pas dans la liste, la valeur est automatiquement ajoutée.

Le moment de génération des fichiers de reprise est {TEMPS_BASE} + {valeur}.

Exemple : backuptimesteps=-600|-120|0

✓ Résultats

- ↳ stationsresultat (obligatoire) : liste des codes sites (8 caractères) et/ou stations (10 caractères) hydro pour lesquelles le PIM doit fournir un résultat, séparés par des « | ». Chaque élément de la liste est un n-uplet dont les champs sont séparés par des « ; » : code site ou station ; code section ; décalage vertical en m. Le numéro de section est utilisé pour lire les fichiers « opt ».
- ↳ stationsresultatcasiers (facultatif) : liste des codes sites (8 caractères) et/ou stations (10 caractères) hydro pour lesquelles le PIM doit fournir un résultat, séparés par des « | ». Chaque élément de la liste est un n-uplet dont les champs sont séparés par des « ; » : code site ou station ; code casier. Le code casier est utilisé pour lire les fichiers « cas_opt ».
- ↳ stationsresultatliaisons (facultatif) : liste des codes sites (8 caractères) et/ou stations (10 caractères) hydro pour lesquelles le PIM doit fournir un résultat, séparés par des « | ». Chaque élément de la liste est un n-uplet dont les champs sont séparés par des « ; » : code station ; code liaison. Le code liaison est utilisé pour lire les fichiers « liai_opt ».

✓ Spécial barrages

- ↳ cheminmascaretbarrage (facultatif) : chemin complet de l'exécutable Mascaret, spécial pour les calculs de barrages.

1.6.3 Formats de fichiers

1.6.3.1 LOI

Le format LOI est un fichier texte, avec séparateur espace, au format suivant : deux colonnes

- ✓ temps : nombre de secondes. Ce nombre peut être négatif.
- ✓ valeur : Hauteur ou Débit selon.

Les trois premières lignes sont un entête ayant pour contenu :

```
# [NOM]
# Temps (s) [TYPE]
S
```

- ✓ [NOM] est le nom de la loi.
- ✓ [TYPE] est « Z » pour des données de type Hauteur, « Q » pour des données de type Débit.

Exemple :

```
# falousevaux
# Temps (s) Q
S
0      2.845
3600   2.845
7200   2.8425
10800  2.84
14400  2.835
18000  2.8325
...
```

1.6.3.2 OPT

Le format OPT est un format texte disposant de deux sections entre crochets (« variables » et « résultats »). La première section décrit les colonnes additionnelles qui sont ajoutées à la seconde section.

Chaque « variable » est une ligne de colonnes séparées par des « ; » :

- ✓ Nom de la variable
- ✓ Code de la variable
- ✓ Unité de la variable
- ✓ Nombre de chiffres après la virgule (entier)

La section résultat contient un tableau, avec séparateur « ; » dont les colonnes sont les suivantes :

- ✓ Temps : en secondes depuis le démarrage de la simulation
- ✓ Numéro de bief : numéro d'ordre
- ✓ Abscisse de la section de calcul : nombre en km (PK)
- ✓ L'identifiant du site (point kilométrique)
- ✓ Les colonnes de variables, dans l'ordre décrit dans la section « variables »

Afin de reconstituer le débit total, il faut additionner les colonnes des variables « Débit mineur » (débit du lit mineur) et « Débit majeur » (débit du lit majeur).

Exemple :

```
[variables]
"Cote du fond";"ZREF";"m";4
"Cote de l'eau";"Z";"m";3
"Debit mineur";"QMIN";"m3/s";3
"Debit majeur";"QMAJ";"m3/s";3
```

```

"Coefficient de frottement mineur";"KMIN";"m1/3/s";0
"Coefficient de frottement majeur";"KMAJ";"m1/3/s";0
"Nombre de Froude";"FR";"";5
"Debit total";"Q";"m3/s";3
[resultats]
  1638000.0;" 5";" 1754"; 78770.00; -3.6600; 2.468;
89.670; 0.000; 53.; 40.; 0.03773; 89.670
  1638000.0;" 2";" 296"; 16071.00; -5.8400; 2.418;
139.770; 0.000; 28.; 28.; 0.01500;
139.770
  1638000.0;" 1";" 10"; 518.00; -11.1100; 2.401;
163.970; 0.000; 30.; 30.; 0.00832;
163.970
  1641600.0;" 5";" 1754"; 78770.00; -3.6600; 2.470;
88.805; 0.000; 53.; 40.; 0.03735; 88.805
  1641600.0;" 2";" 296"; 16071.00; -5.8400; 2.970;
-656.587; 0.000; 28.; 28.; -0.06200; -
656.587
  1641600.0;" 1";" 10"; 518.00; -11.1100; 3.401;
-1398.448; 0.000; 30.; 30.; -0.06264; -
1398.448
  1645200.0;" 5";" 1754"; 78770.00; -3.6600; 3.265;
58.940; 0.000; 53.; 40.; 0.01892; 58.940

```

1.6.3.3 Fichier résultats des casiers « .casiers_opt »

Le format est similaire au format « OPT ».

La différence est dans le nombre de colonnes fixes pour la section résultat :

- ✓ Temps : en secondes depuis le démarrage de la simulation
- ✓ Nom du casier
- ✓ Les colonnes de variables, dans l'ordre décrit dans la section « variables »

```

/* resultats casiers
[variables]
"cote surface libre casier";"ZCAS";"m";2
"surface casier";"SURCAS";"m2";2
"volume casier";"VOLCAS";"m3";2
[resultats]
  0.;"CAS 1"; 23.00; 360400.41; 0.00
  0.;"CAS 2"; 23.00; 495097.52; 0.00
  0.;"CAS 3"; 22.00; 945193.90; 0.00
  0.;"CAS 4"; 22.00; 1148572.83; 0.00
  0.;"CAS 5"; 21.01; 1609692.16; 16017.23
  0.;"CAS 6"; 23.00; 1275179.10; 0.00
  0.;"CAS 7"; 19.00; 653103.08; 0.00
  0.;"CAS 8"; 19.00; 833053.36; 0.00
  0.;"CAS 9"; 19.00; 1935972.17; 0.00
  0.;"CAS 10"; 16.00; 1486087.32; 0.00
  0.;"CAS 11"; 17.00; 685717.67; 0.00
  0.;"CAS 12"; 16.00; 2352004.08; 0.00
  0.;"CAS 13"; 15.01; 486534.20; 4825.46
  0.;"CAS 14"; 14.00; 332503.36; 0.00
  0.;"CAS 15"; 13.00; 1015893.76; 0.00
3600.;"CAS 1"; 23.00; 360400.41; 0.00
3600.;"CAS 2"; 23.00; 495097.52; 0.00
3600.;"CAS 3"; 22.00; 945193.90; 0.00

```

3600.; "CAS 4";	22.00;	1148572.83;	0.00
3600.; "CAS 5";	21.01;	1609692.16;	16017.23
3600.; "CAS 6";	23.00;	1275179.10;	0.00
3600.; "CAS 7";	19.00;	653103.08;	0.00
3600.; "CAS 8";	19.00;	833053.36;	0.00
3600.; "CAS 9";	19.00;	1935972.17;	0.00
3600.; "CAS 10";	16.00;	1486087.32;	0.00
3600.; "CAS 11";	17.00;	685717.67;	0.00
3600.; "CAS 12";	16.00;	2352004.08;	0.00
3600.; "CAS 13";	15.01;	486534.20;	4825.46
3600.; "CAS 14";	14.00;	332503.36;	0.00
3600.; "CAS 15";	13.00;	1015893.76;	0.00

1.6.3.4 Fichier archive des casiers « .casiers_pim »

Ce type de fichier est généré par le PIM, pour sauvegarder des résultats des casiers à un instant de calcul. Son contenu est obtenu à partir d'un fichier résultat « .casiers_opt » pour un temps donné.

Ce fichier ne contient qu'une liste de valeurs numériques, une valeur par ligne.

Par exemple avec un résultats casiers, fichier « .casiers_opt », pour le temps « 3600 » :

3600.; "CAS 1";	23.00;	360400.41;	0.00
3600.; "CAS 2";	23.00;	495097.52;	0.00
3600.; "CAS 3";	22.00;	945193.90;	0.00
3600.; "CAS 4";	22.00;	1148572.83;	0.00
3600.; "CAS 5";	21.01;	1609692.16;	16017.23
3600.; "CAS 6";	23.00;	1275179.10;	0.00
3600.; "CAS 7";	19.00;	653103.08;	0.00
3600.; "CAS 8";	19.00;	833053.36;	0.00
3600.; "CAS 9";	19.00;	1935972.17;	0.00
3600.; "CAS 10";	16.00;	1486087.32;	0.00
3600.; "CAS 11";	17.00;	685717.67;	0.00
3600.; "CAS 12";	16.00;	2352004.08;	0.00
3600.; "CAS 13";	15.01;	486534.20;	4825.46
3600.; "CAS 14";	14.00;	332503.36;	0.00
3600.; "CAS 15";	13.00;	1015893.76;	0.00

le fichier « .casiers_pim » correspondant sera:

23.00
23.00
22.00
22.00
21.01
23.00
19.00
19.00
19.00
16.00
17.00
16.00
15.01
14.00
13.00

1.6.3.5 Fichier des cotes de crêtes « Fichier_Crete.csv »

Ce type de fichier est généré par Mascaret, il est spécifique au run spécial barrage de Mascaret.

Voici un exemple de contenu :

2		
0.00	168.00	barrage_drace
0.00	166.00	barrage_couzon
3600.00	168.10	barrage_drace
3600.00	165.90	barrage_couzon
7200.00	168.20	barrage_drace
7200.00	165.80	barrage_couzon
10800.00	168.30	barrage_drace
10800.00	165.70	barrage_couzon
14400.00	168.40	barrage_drace
14400.00	165.60	barrage_couzon
18000.00	168.50	barrage_drace
18000.00	165.50	barrage_couzon
21600.00	168.60	barrage_drace
21600.00	165.42	barrage_couzon
25200.00	168.70	barrage_drace
25200.00	165.42	barrage_couzon
28800.00	168.80	barrage_drace

1.6.3.6 Fichier archive des cotes de crêtes « .barrages_pim »

Ce type de fichier est généré par le PIM, pour sauvegarder les cotes de crêtes pour chaque barrage. Par barrage, il s'agit d'une ligne extraite du fichier « Fichier_Crete.csv ».

Voici un exemple de contenu :

172800.00	168.81	barrage_drace
172800.00	165.70	barrage_couzon

Le pattern pour le nom de ces fichiers est le suivant : « {0}_{1}_{2}_{3}.barrages_pim » avec pour les champs successifs :

- ✓ l'horodate au format YYYYMMDD_hh_mm
- ✓ la session courante, « SESSION_TR » ou le code d'un session temps différé
- ✓ le code du modèle et le code du scénario
- ✓ le type de run

Voici un exemple :

« 20191126_0000_SESSION_TR_00yMAS0221_Sp01_Arpege_Dryrun-REAL_TIME.barrages_pim »

1.7 Arborescence générale

La machine virtuelle Mascaret est organisée comme suit :

- ✓ Répertoires des modèles Mascaret (cf. 1.7.1.1) :

/home/mascaret/pom/modeles

- ✓ Répertoires d'échange POM :

```
/home/mascaret/pom/echanges
```

✓ Répertoires du PIM (cf. 1.7.2) :

```
/home/mascaret/pom/pim
```

1.7.1 Arborescences des fichiers Mascaret

Ce chapitre présente l'arborescence des fichiers nécessaires à un lancement de Mascaret.

1.7.1.1 Arborescence d'un modèle Mascaret

Le répertoire des modèles contient un sous-répertoire par modèle :

```
/home/mascaret
|- pom
   |- modeles
      |- {CODE_MODELE1}
         |- FichierCas.txt
         |- ...
      |- {CODE_MODELE2}
         | FichierCas.txt
         |- ...
   ...
```

Chaque répertoire de modèle contient les fichiers et répertoires nécessaires au lancement de Mascaret.

Il contient obligatoirement un fichier « FichierCas.txt » directement dans le répertoire. Ce fichier a pour contenu uniquement le chemin relatif du fichier « xcas » du modèle.

Ce fichier « xcas » est au format « xml » et définit l'architecture du modèle, notamment les chemins relatifs de tous les autres fichiers du modèle :

- ↳ le fichier de géométrie (« .geo »)
- ↳ les éventuels fichiers de tarage (« .loi ») et autres lois statiques (seuils...)
- ↳ les fichiers de ligne d'eau (« .lig »)
- ↳ les éventuels fichiers de casiers (« .casier »)
- ↳ le fichier de log du calcul (« .lis »)
- ↳ le fichier des résultats finaux (« .opt ») (à lire par le PIM), cf. 1.6.3.2

Exemple d'un dossier modèle simple :

```
/home/mascaret/pom/modeles/{CODE_MODELE}
|- FichierCas.txt
|- apport_PK33.loi
|- apport_PK58.loi
|- CT_LaReole.loi
|- geometrie.geo
|- LigneEauInitiale.lig
|- ParametresMascaret.xcas
|- ResultatsListing.lis
|- ResultatsOpthyca.opt
```

Dans ce cas, le contenu de FichierCas.txt est :

```
./ParametresMascaret.xcas
```

Exemple d'un dossier modèle avec un sous-répertoire :

```
/home/mascaret/pom/modeles/{CODE_MODELE}
|- FichierCas.txt
|- DonneesDyn
   |- apport_PK33.loi
   |- apport_PK58.loi
|- DonneesStat
   |- CT_LaReole.loi
   |- geometrie.geo
   |- LigneEauInitiale.lig
   |- ParametresMascaret.xcas
|- Resultats
   |- ResultatsListing.lis
   |- ResultatsOpthyca.opt
```

Dans ce cas, le contenu de FichierCas.txt est :

```
./DonneesStat/ParametresMascaret.xcas
```

1.7.2 Arborescence de travail du PIM

Afin de lancer des calculs indépendants les uns des autres, le PIM s'exécute dans une arborescence qui lui est propre :

```
/home/mascaret/pom/pim
|- workspace (répertoire de travail PIM)
   |- archives
      |- {CODE_MODEL_MASCARET}
         |- {TPS_BASE}__{TYPE_SESSION}__{SCENAR_POM}__{MODE}.pim
         |- {TPS_BASE}__{TYPE_SESSION}__{SCENAR_POM}__{MODE}.barrages_pim
      |- {CODE_SESSION_POM}
         |- {DATE_PIVOT}
            |- {MODE_CALCUL_POM}
               |- {CODE_SCENARIO_POM}
                  |- {CODE_MODELE_MASCARET}
                     |- {MODE_CALCUL_MASCARET}
                        |- analyse
                           |- FichierCas.txt
                           |- ...
                        |- {NOM_SERIE_1}
                           |- FichierCas.txt
                           |- ...
                        |- {NOM_SERIE_2}
                           |- FichierCas.txt
                           |- ...
                     ...
            ...
         ...
      ...
   ...
```

Les conventions suivantes sont prises :

- ✓ Les dates sont au format « AAAAMMDD_HHMM » pour effectuer un tri alphabétique chronologique
- ✓ Le mode de calcul POM est lu dans la balise « session → mode » du fichier « parameters.xml ». Il vaut « REAL_TIME » s'il n'est pas renseigné.
- ✓ Le répertoire {CODE_MODELE_MASCARET} est le nom du « **répertoire du modèle Mascaret** » à lancer (clef [models]names du fichier « .ini »).
- ✓ Le répertoire {MODE_CALCUL_MASCARET} correspond au mode de calcul Mascaret lu dans le mode de calcul du scénario POM.
- ✓ Les répertoires « analyse » et « {NOM_SERIE_i} » (pour les runs de prévision) correspondent aux différents runs à lancer dans Mascaret. Ils contiennent les fichiers nécessaires au calcul (cf. 1.9)

1.7.3 Nomenclatures

Par la suite, les dénominations suivantes sont prises :

- ✓ le répertoire ci-dessous est appelé « **répertoire du modèle Mascaret** »

```
/home/mascaret/pom/modeles/{NOM_MODELE}
```

- ✓ Le répertoire ci-dessous est appelé « **répertoire de travail PIM** »

```
/home/mascaret/pom/pim/workspace/{CODE_SESSION_POM}/{DATE_PIVOT}/  
{MODE_CALCUL_POM}/{CODE_SCENARIO_POM}/{CODE_MODELE_MASCARET}
```

- ✓ le répertoire ci-dessous est appelé « **répertoire de travail Mascaret** »

```
{répertoire de travail PIM}/{MODE_CALCUL_MASCARET}/analyse/
```

ou

```
{répertoire de travail PIM}/{MODE_CALCUL_MASCARET}/{NOM_SERIE}/
```

1.8 Installation

A partir du PIM 3.2, l'OS d'installation est Linux Debian 10. Il s'agit par défaut d'une VM fournie au format VMDK.

Pour l'installation, le répertoire suivant permet de déposer les fichiers nécessaires au PIM :

```
/home/mascaret/pom/pim/install
```

Il s'agit des fichiers :

- ✓ d'installation du PI : **pominterface-4.0.X-py3-none-any.whl**
- ✓ d'installation du PIM : **pim-4.0.X-py3-none-any.whl**

Le PIM est prévu pour être installé de manière standard avec l'outil « pip ».

Exemple de commande d'installation :

```
$ pip install pim-4.0.X-py3-none-any.whl
```

1.9 Lancement de Mascaret

1.9.1 Modes de calculs de Mascaret

Mascaret est un modèle hydraulique sans possibilité native d'assimilation.

Le PIM utilise le mode « DRYRUN » pour lancer des calculs de Mascaret sans assimilation.

Le paramétrage d'un calcul de Mascaret s'effectue dans un fichier nommé « ParametresMascaret.xcas ».

Afin de permettre au PIM d'exploiter plusieurs formats de fichier « xcas », les deux balises suivantes sont autorisées :

- ✓ <fichierCas>
- ✓ <fichier_cas>

1.9.2 Régimes Mascaret

Les différents régimes de calcul de Mascaret sont les suivants (ils correspondent à des cœurs de calculs différents) :

- ✓ code 1 = SARAP (écoulement fluvial permanent sur un réseau ramifié), pour l'initialisation des calculs
- ✓ code 2 = REZO (écoulement fluvial transitoire sur un réseau ramifié ou maillé),
- ✓ code 3 = MASCARET (écoulement transitoire transcritique sur un réseau ramifié et maillé)

Ils sont paramétrés dans le fichier « .xcas » comme suit :

```
<fichierCas>
  <parametresCas>
    <parametresGeneraux>
      <versionCode>3</versionCode>
      <code>3</code>
      <fichMotsCles>mascaret0.cas</fichMotsCles>
      <dictionnaire>dico.txt</dictionnaire>
      <progPrincipal>princi.f</progPrincipal>
      <sauveModele>>false</sauveModele>
      <fichSauvModele>mascaret0.tmp</fichSauvModele>
      <validationCode>>false</validationCode>
      <typeValidation>1</typeValidation>
      <presenceCasiers>>false</presenceCasiers>
      <bibliotheques>
        <bibliotheque>mascaretV5P1.a damoV3P0.a</bibliotheque>
      </bibliotheques>
    </parametresGeneraux>
  ...
```

Chaque calcul doit donc renseigner le code du régime Mascaret à utiliser. Le régime par défaut à utiliser est indiqué dans le fichier « xcas » du modèle dans son répertoire de configuration par défaut (cf. § 1.7.1.1 - Arborescence d'un modèle Mascaret).

2. Adaptations du PI v3

L'architecture logicielle est basée sur le PI avec surcharge de processeurs ou fonctions par le PIM.

Les points ci-dessous précisent les fonctionnalités du PI à utiliser.

2.1 Mode de calcul

Le mode de calcul ne comporte aucun mot clef supplémentaire à ceux gérés dans le PI.

Pour le mot clé {ANALYSE}, le PIM applique les vérifications suivantes :

- ↳ Le scénario d'analyse renseigné doit donner lieu à un run d'analyse
- ↳ Le mode de lancement de Mascaret (DRYRUN) de ce scénario doit être le même que le mode Mascaret du scénario d'analyse renseigné.
- ↳ Si ce paramètre est renseigné sur un scénario, il doit donner lieu à un run de prévision (au moins)

2.2 Nettoyage des fichiers

L'objectif du nettoyage est d'éviter d'écraser des fichiers utilisés dans des runs en cours ou bien de réutiliser par mégarde des fichiers issus d'un run précédent. Pour un scénario POM, dans l'arborescence prévue (cf. §1.7.2 - Arborescence de travail du PIM), cela correspond aux répertoires suivants :

```
/home/mascaret/pom/pim/workspace/{CODE_SESSION_POM}/{DATE_PIVOT}/
{MODE_CALCUL_POM}/{CODE_SCENARIO_POM}/{CODE_MODELE_MASCARET}/
{MODE_CALCUL_MASCARET}
|- analyse
|- {NOM_SERIE_1}
|- {NOM_SERIE_2}
...
```

En l'état les processeurs ne permettent pas de supprimer aussi finement les fichiers. Il faut donc supprimer tous les fichiers d'échange et de travail plus âgés qu'un jour ce qui correspond à une implémentation du nettoyage comme suit :

```
[cleaning]
cleaners=echanges,workspace

[echanges]
directory=/home/mascaret/pom/echanges/*
age=1
filter=*
mode=prerun

[workspace]
directory=/home/mascaret/pom/pim/workspace/*
age=1
filter=*
mode=prerun
```

2.3 Calculs en parallèle

Le PIM peut s'exécuter plusieurs fois simultanément et est paramétré en ce sens.

A chaque lancement, un fichier de verrou est créé dans le « **répertoire de travail PIM** » (cf §1.7.3 - Nomenclatures) afin qu'aucun autre lancement du PIM ne puisse venir le concurrencer.

2.4 Un « run » par scénario

Le PI fonctionne différemment selon qu'il lance (au moins) un « run » par scénario POM ou un unique « run » pour tous les scénarios.

Dans le cas du PIM, chaque scénario POM donne lieu à au moins un « run ».

2.5 Gestion de l'avancement

L'avancement est géré de manière générique comme suit :

- ✓ 5% : lancement de PIM
- ✓ X% : run
- ✓ 95% : finalisation
- ✓ 100% : fin de PIM

La phase « run » représente donc 90 %.

Chaque scénario est représenté à part égale dans ces 90 %. Chaque scénario occupe donc 90/N % (N = nombre de scénarios) et son avancement se décompose comme suit :

- ✓ 20 % : lecture des entrées (chaque entrée occupe 20/N %, où N est le nombre d'entrée)
- ✓ 60 % : lancements de l'exécutable Mascaret
- ✓ 20 % : traitement des sorties (chaque sortie occupe 20/N %, où N est le nombre de sorties)

2.6 Gestion des unités

Les valeurs des données dans Mascaret sont dans les unités suivantes :

- ✓ en « m » pour les hauteurs
- ✓ « m³/s » pour les débits

Les données étant fournies au format XML Sandre, des conversions systématiques sont à prévoir :

- ✓ en « mm » pour les hauteurs
- ✓ en « l/s » pour les débits

2.7 Dates des simulations analyse/prévision

Le PIM fonctionne en ANALYSE / PREVISION.

Les données sont séparées en 2 paquets :

- Analyse entre T0 et T1
- Prévision entre T1 et T2

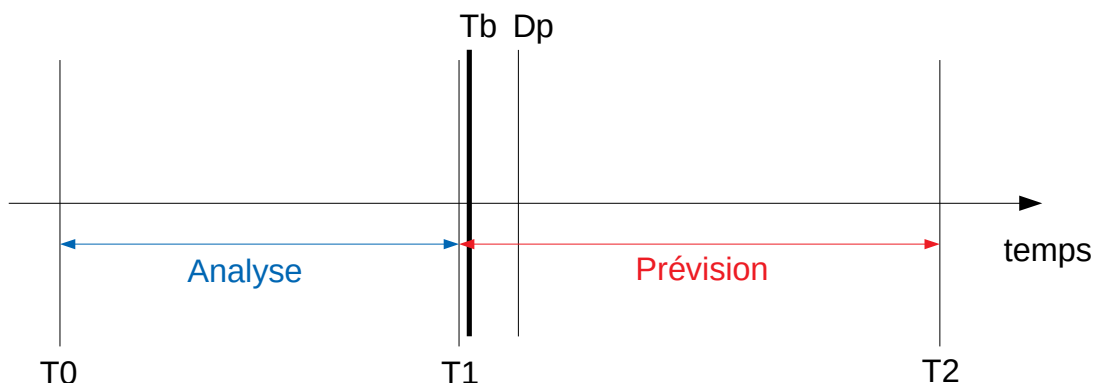


Figure 2 : dates des simulations

Les dates fournies par la POM sont en TU alors que Mascaret fonctionne en temps relatif en secondes. Il commence classiquement à 0 en début de simulation (la date de début de la simulation). Ainsi, toutes les dates POM sont reliées au temps Mascaret par l'équation ci-dessous :

$$\text{Temps Mascaret} = \{\{ \text{DATE_DONNEE} \}\} - \{\{ \text{DATE_DEBUT_SIMULATION} \}\}$$

Les dates absolues sont converties en temps relatif du modèle. Il s'agit d'un nombre de secondes, partant de l'instant $\{\{ \text{DATE_DEBUT_SIMULATION} \}\}$ (pour lequel le temps relatif vaut 0). Une fois ce $\{\{ \text{DATE_DEBUT_SIMULATION} \}\}$ déterminé, toutes les dates absolues peuvent être converties en temps relatif et vice versa.

Pour l'Analyse :

- ✓ $\{\{ \text{DATE_DEBUT_SIMULATION} \}\} = T0$
- ✓ $\{\{ \text{DATE_FIN_SIMULATION} \}\} = T1$

Pour la Prévision :

- ✓ $\{\{ \text{DATE_DEBUT_SIMULATION} \}\} = T1$
- ✓ $\{\{ \text{DATE_FIN_SIMULATION} \}\} = T2$

2.8 Méthodes de démarrage de l'analyse à T0

2.8.1 Méthodes d'initialisation gérées par le PIM

Le PIM utilise les méthodes d'initialisation suivantes définies dans le PI3 :

- ✓ REPRISE
- ✓ DEFAUT

Si le PIM rencontre une méthode d'init « AUCUNE » ou « CONSTANTE », cela génère une erreur bloquante.

Le PIM définit une méthode d'initialisation « PERMANENT ».

Ce type d'initialisation vise à lancer un run spécifique (dit « permanent ») censé générer un fichier de reprise utilisable pour le véritable run d'analyse (dit « non permanent »).

Ce mode d'initialisation aboutit donc à deux runs configurés différemment.

La méthode « PERMANENT » est implémentée comme indiquée ci-dessous.

2.8.2 Mode d'initialisation « PERMANENT »

Dans le cas d'une initialisation « PERMANENT », deux runs sont lancés consécutivement :

- ✓ Un run « PERMANENT » : lancé en mode 1 (SARAP) sur une plage de dates [0, 3600], avec la même valeur pour ces deux dates dans les fichiers « .loi », à savoir la valeur observée à T0.

Ce délai de 3600 n'est pas paramétrable.

Le lancement d'un run « PERMANENT » ne nécessite pas de ligne d'eau initiale.

Note : Dans la version v8p2r0 du code Mascaret, les modèles Mascaret qui utilisent une courbe de tarage en condition limite ne peuvent pas être lancés en permanent et cette option est donc à exclure pour ces modèles .

- ✓ Un run d'analyse (non permanent), initialisé en reprise à partir du fichier produit par le run précédent, dans le régime Mascaret prévu par défaut dans le fichier « .xcas » (2=REZO ou 3=MASCARET)

Ce run d'analyse est lancé sur une plage [T0 ; T1].

2.8.3 Mode d'initialisation «REPRISE»

En mode « REPRISE » (sur une archive), le PIM renseigne la balise « parametresConditionsInitiales » du fichier « ParametresMascaret.xcas » et notamment :

- ↳ « parametresConditionsInitiales → ligneEau → LigEauInit » à « True »
- ↳ les informations du fichier de ligne d'eau issu de l'archive (chemin complet)

```
<parametresConditionsInitiales>
  <repriseEtude>
    <repriseCalcul>true</repriseCalcul>
  </repriseEtude>
  <ligneEau>
    <LigEauInit>true</LigEauInit>
```

```

<modeEntree>1</modeEntree>
<fichLigEau>Archive/mascarethorodatage.lig</fichLigEau>
<formatFichLig>2</formatFichLig>
<nbPts>0</nbPts>
</ligneEau>
</parametresConditionsInitiales>

```

Lors du lancement de Mascaret par le PIM via l'API Python, la valeur de la propriété «parametresConditionsInitiales | ligneEau | fichLigEau » est utilisée :

```

# INIT HYDRO
masc.init_hydro_from_file("Archive/mascarethorodatage.lig")

```

2.8.4 Mode d'initialisation « DEFAULT »

En mode « DEFAULT », le PIM renseigne la balise « parametresConditionsInitiales » du fichier « ParametresMascaret.xcas » et notamment :

- ↳ « parametresConditionsInitiales → ligneEau → ligEauInit » à « true »
- ↳ les informations du fichier de ligne d'eau : un des fichiers .lig du dossier « default »

```

<parametresConditionsInitiales>
  <repriseEtude>
    <repriseCalcul>true</repriseCalcul>
  </repriseEtude>
  <ligneEau>
    <LigEauInit>true</LigEauInit>
    <modeEntree>1</modeEntree>
    <fichLigEau>Default/LigneEau_*.lig</fichLigEau>
    <formatFichLig>2</formatFichLig>
    <nbPts>0</nbPts>
  </ligneEau>
</parametresConditionsInitiales>

```

Le PIM doit définir le fichier ligne d'eau dans le script de lancement de Mascaret via l'API Python :

```

# INIT HYDRO
masc.init_hydro_from_file("Default/LigneEau_*.lig")

```

2.8.4.1 Initialisation RAMPE

Le principe cette initialisation est d'utiliser un débit ou une hauteur à une échéance avant la première observation. Cela permet de démarrer la simulation avec des conditions limites cohérentes avec la ligne d'eau (en particulier lorsque les premières observations sont éloignées des conditions limites qui ont permis d'obtenir la ligne d'eau utilisée dans le mode DEFAULT) et de rejoindre progressivement le véritable état initial des premières observations sans provoquer de choc numérique.

Ce type d'initialisation est possible uniquement avec la méthode d'initialisation « DEFAULT ».

La définition de l'initialisation RAMPE se fait dans le fichier .ini de la manière suivante :

- ➔ une section **[init-rampe]**
 - ↳ dont la seule clé « **rampe-sections** » liste d'autres sections du fichier .ini
- ➔ chaque section définie dans « **rampe-sections** »
 - ↳ doit être définie, sinon une erreur de lecture est levée
 - ↳ contient une clé « **file** » : le nom du fichier ligne d'eau à utiliser qui doit être dans le répertoire « backupdirectory » pour le mode « DEFAULT »
 - ↳ contient une clé « **time** » : le temps d'initialisation avant T0 de début de calcul, l'unité est la seconde.
La valeur est toujours ≥ 0 .
Une valeur positive indique un temps orienté vers le passé à partir de T0.
La valeur « 0 » indique que cette « rampe-sections » est inactive.
Une valeur négative génère une erreur de lecture.
 - ↳ contient une clé « **stations** » : une liste de tuples séparés par des « | », chaque tuple est un code de site ou station et une valeur, séparés par « ; », la valeur est en m³/s pour des débits ou en m pour des hauteurs

Voici un exemple de contenu :

```
[init-rampe]
rampe-sections= rampe1, rampe2, rampe-test

[rampe1]
file = LigneEauInitiale_1.lig
time = 86400
stations = H5172010;60|H5091010;10|H520100501;3.6

[rampe2]
file = LigneEauInitiale_2.lig
time = 3600
stations = H5172010;30|H5091010;5|H520100501;3.6

[rampe-test]
file = LigneEauInitiale_test.lig
time = 3600
stations = H5172010;10|H5091010;1.0|H520100501;1.0
```

3. Périmètre fonctionnel du PIM

Le présent chapitre décrit les étapes à mener pour le lancement des calculs Mascaret. Il détaille les étapes à modifier par rapport au PI.

3.1 Lancement

Le lancement du PIM est identique au lancement du PI : mêmes arguments de ligne de commande, avec la possibilité facultative de renseigner un fichier « .ini ».

Note : le nom du fichier « .ini » utilisé par défaut est « pim.ini ».

3.2 Ordonnanceur de tâches

L'ordonnanceur hérite de la classe « ModelBase » et s'appelle « ModelMascaret ».

La méthode principale d'ordonnement n'est pas modifiée, elle reprend donc les étapes du PI :

- ✓ Initialisation
- ✓ Nettoyage pré run
- ✓ Traitement des scénarios
- ✓ Nettoyage post run

Comme indiqué ci-avant, le PIM est paramétré pour lancer au moins un run par scénario : le lancement éventuel de run après le traitement des scénarios (fonctionnalité utilisée dans le PIG) n'est donc pas effectué.

3.3 Initialisation

L'initialisation du PIM est identique à celle du PI aux points ci-dessous près :

- ✓ la vérification du fichier « parameters.xml » fourni par la POM doit être enrichie :
 - ↳ Il ne peut y avoir qu'une ou aucune ressource de type « fichier » en sortie.
 - ↳ Il ne peut y avoir que des grandeurs « H » ou « Q » en entrée (quel que soit le scénario)
- ✓ La lecture du fichier « .ini » doit être adaptée pour lire les paramètres spécifiques au PIM (cf. 1.6.2).
- ✓ Vérification des modes de calcul des scénarios
- ✓ Le scénario principal doit donner lieu à un run d'analyse (i.e. contenir des données avant le temps de base)

3.4 Traitement des scénarios

A partir de la version 3.x, le PIM gère plusieurs modèles, comme prévu par le PI (via la section du PI « [models] »).

3.4.1 Run(s) d'analyse

Pour le(s) run(s) d'analyse, on va tenter de trouver la bonne méthode d'initialisation en mode DRYRUN. Pour cela, on boucle sur les méthodes d'initialisation jusqu'à ce qu'une fonctionne pour ce mode. Pour chaque méthode :

- ↳ Si la méthode est « PERMANENT », il faut lancer un run permanent puis un run en reprise sur le résultat du run permanent (cf 2.8).
- ↳ Sinon, pour les autres modes, il faut lancer un run seul dont l'initialisation dépend de la méthode en cours.

Comme défini dans le PI, le(s) run(s) d'analyse est (sont) lancé(s) sur une plage [T0, T1] (cf specs PI3, § 2.4).

Avant chaque nouvelle tentative de run d'analyse, le dossier de calcul « analyse » est vidé.

Voici le tableau récapitulatif des lancements possibles pour les runs d'analyse :

	DEFAUT	REPRISE	PERMANENT
DRYRUN	Calcul 1 : init : DEFAUT mode : DRYRUN fichier reprise : défini via '.ini' dans <backupdirectory>	Calcul 1 : init : REPRISE mode : DRYRUN fichier reprise : cherche dans 'archives'	Calcul 1 : init : PERMANENT mode : DRYRUN Calcul 2 : init : REPRISE mode : DRYRUN fichier reprise : résultat calcul 1
	Résultat Analyse : fichiers .lig, .opt du calcul 1	Résultat Analyse : fichiers .lig, .opt du calcul 1	Résultat Analyse : fichiers .lig, .opt du calcul 2

Tableau 2 : lancement des runs d'analyse

3.4.2 Run(s) de prévision

Le run de prévision est un run unique (quel que soit le mode de calcul Mascaret).

Pour constituer le(s) run(s) de prévision, le PIM boucle sur les séries à propager.

Pour chacune, le PIM lance un run sur les données de la série courante, initialisé en reprise sur le résultat du run d'analyse. Le run d'analyse est issu par ordre de priorité :

- ✓ du mot clef « analyse » du mode de calcul POM, s'il est renseigné (cf. 2.1),
- ✓ du scénario courant s'il a donné lieu à un run d'analyse,

- ✓ du scénario principal.

Il y a autant de runs de prévisions que de séries à propager.

Les données d'entrée d'un run de prévision sont :

- ✓ la ligne d'eau produite par le run d'analyse (relay_prevision)¹,
- ✓ les fichiers lois générés sur les données d'entrées fournies par la POM.
- ✓

Comme défini dans le PI, le(s) run(s) de prévision est (sont) lancé(s) sur une plage [T1, T2] (cf specs PI3, § 2.4).

Voici le tableau récapitulatif des lancements possibles pour les runs de prévision :

	Prévision
DRYRUN	<p>Recherche calcul Analyse avec mode DRYRUN</p> <p>Calcul 1 : init : REPRISE mode : DRYRUN fichier reprise : résultat calcul Analyse</p> <p>Résultat Prevision : fichier .opt du calcul 1</p>

Tableau 3 : lancement des runs de prévision

3.5 Traitement des entrées

La lecture des entrées est réalisée comme dans le PI : toutes les données sont chargées dans une liste commune.

Pour chaque scénario, on ne lit que les données du scénario courant.

3.6 Gestion des Runs

Le lancement des runs de Mascaret suit la procédure du PI. Les chapitres ci-dessous précisent les fonctions à surcharger dans les Plx.

Pour rappel, les runs sont liés au scénario POM en cours de traitement.

Le PIM gère 2 types de run : un run standard et un run spécial barrages.

Le PIM ne lance le run spécial barrages que si le fichier nommé « Fichier_Barrage_Mobile.txt » est présent dans le dossier modèle mascaret.

¹ cf. specification PI, paragraphe 2.4.3 Démarrage d'un calcul de prévision (à T1), association runs d'analyse/runs de prévision

3.6.1 LaunchProcessor : initialisation RAMPE

Dans le PI, la méthode d'init « DEFAULT » consiste à créer une boucle de lancement de calculs sur chacun des fichiers ligne d'eau listés dans l'option « init-files ».

Pour prendre en compte l'initialisation RAMPE, le PIM modifie ce fonctionnement lors de la construction de la boucle de lancement des calculs :

- ✓ chercher la clé « rampe-sections » correspondant au fichier ligne d'eau en cours : il faut que la clé « file » corresponde au nom du fichier ligne d'eau de la boucle et que la valeur « time » soit > 0
- ✓ si la clé « rampe-sections » est trouvée, le PIM va activer l'option « init_rampe » pour ce lancement de calcul et associer la « rampe-sections » trouvée au paramétrage de ce lancement de calcul.

3.6.2 Types de run

Chaque scénario POM donne lieu à différents types de runs :

- ✓ un run d'analyse sans erreur (plusieurs runs sont envisageables selon les tentatives d'initialisation, mais un seul, le dernier, doit aboutir) si le scénario POM contient des données situées avant le temps de base
- ✓ un ou plusieurs runs de prévision (propagation des séries, ...) si le scénario contient des données après le temps de base. Le type de run Mascaret est DRYRUN.

3.6.3 Copie des fichiers du modèle

Le PIM copie les fichiers de paramétrage du modèle depuis le répertoire (cf. 1.7.1)

```
/home/mascaret/pom/modeles/{MODELE}
```

vers le « **répertoire de travail Mascaret** » (cf. 1.7.3)

```
{répertoire de travail PIM}/{MODE_CALCUL_MASCARET}/analyse
```

ou

```
{répertoire de travail PIM}/{MODE_CALCUL_MASCARET}/{NOM_SERIE}
```

selon la nature du run analyse ou prévision.

Le PIM copie tous les fichiers et sous-dossiers sauf les fichiers liés aux résultats : les fichiers ayant pour extension ne sont pas copiés :

- ✓ « .opt », « cas_opt », « liai_opt »
- ✓ « .lis », « .cas_lis », « .liai_lis ».

3.6.4 Paramétrage de la plateforme

Le lancement de Mascaret par le PIM nécessite la modification du fichier « ParametresMascaret.xcas » et la génération des fichiers d'entrée « .loi ».

3.6.4.1 Décalage vertical sur les données d'entrées

Le PIM permet d'effectuer une conversion d'échelle sur les hauteurs en entrée.

Pour cela, dans le paramètre **stationsloi**, pour chaque tuple, la troisième valeur correspond au décalage à appliquer en m : « code station; code section ; décalage vertical en m »

Le paramètre décalage vertical n'a pas de sens pour les sites. Un message d'avertissement dans « progression.xml » et dans « pom.log » signale l'incohérence : « Pour le site xxxx, la conversion d'échelle ne peut pas être appliquée sur les entrées ». Ce message n'est pas bloquant, les calculs se poursuivent.

Le PIM ne contrôle pas que pour une station donnée, il y a bien un décalage sur les entrées, paramètre **stationsloi**, et sur les prévisions, paramètre **stationsresultat**.

Pour chaque décalage en entrée qui est appliqué sur une station et une hauteur, un message d'information dans « progression.xml » et dans « pom.log » est tracé : « Pour la station xxx, la conversion d'échelle est appliquée sur les données en entrée avec le décalage vvvvv m », un seul message pour toute la série (ie pas de message pour chaque date)

Pour chaque donnée d'entrée de la station, le décalage en m est ajouté à la donnée.

3.6.4.2 Génération des fichiers « .loi »

Note : le format des fichiers « LOI » est décrit en 1.6.3.1.

Rappel : pour la POM, le symbole de la grandeur « hauteur » est H. Pour Mascaret c'est « Z ».

ATTENTION au cas particulier des runs en régime permanent pour l'initialisation.

Pour connaître le nombre de fichiers à générer il faut récupérer le paramètre « stationsloi » du fichier « .ini » (cf. 1.6.2). Il contient les lois dynamiques dont les données sont fournies par la POM. D'autres lois peuvent exister : elles sont dites statiques (seuil, tarage, ...) et ne sont pas modifiées par le PIM. Toutes les lois sont listées dans la balise « parametresLoisHydrauliques → nb » dans le fichier « ParametresMascaret.xcas ». Il correspond au nombre de balises des stations loi « parametresLoisHydrauliques → lois → structureParametresLoi ». Parmi ces balises, seules les balises correspondant aux lois renseignées dans le fichier « .ini » sont à mettre à jour.

Pour chaque balise de station loi, numéro N (de 1 au nombre de loi) :

- ✓ Récupérer le code Mascaret de la station à générer à partir de la balise « nom » du fichier ParametresMascaret.xcas. On modifie cette balise uniquement si son code apparaît dans le paramètre « stationsloi » du fichier « .ini » (cf. 1.6.2). Dans le cas contraire, on la saute et on passe à la suivante.

ATTENTION : le nom de la station dans le fichier « .xcas » doit être le code entité POM.

- ✓ Calculer la grandeur associée à cette entité : si la balise « type » vaut 1, la grandeur est « débit », sinon c'est « hauteur ».
- ✓ Récupérer le nom du fichier à générer dans la balise « donnees → fichier »
- ✓ Filtrer les données d'entrées sur le code entité et la grandeur calculée
- ✓ Trier les données par ordre croissant de date
- ✓
- ✓ Pour chaque donnée
 - ↳ Tester si sa date est dans la plage des données utilisées pour ce run : avant le temps de base pour un run d'analyse, après le temps de base pour un run de prévision. Si c'est le cas, l'ajouter en :
 - prenant la valeur de la série à traiter : valeur moyenne pour un run d'analyse, sinon, valeur de la série en cours pour un run de prévision (propagation des séries). Si cette série n'existe pas, la valeur moyenne est utilisée.

- en calculant pour chaque date le temps relatif par rapport à DATE_DEBUT_SIMULATION.
- en modifiant les unités : les valeurs sont en « m » pour les hauteurs, en « m³/s » pour les débits (valeurs à convertir car le XML sandre contient des hauteurs en « mm » et des débits en « l/s »)
- pour les hauteurs, prise en compte du « décalage en m » issu de « stationsloi »
- ✓ Ajouter en cas de besoin des données : l'objectif est de prolonger la plage de données pour qu'elle occupe toute la plage de calcul (sinon Mascaret plante) :
 - Si la date de la première donnée est supérieure à la date de début du run, ajouter une donnée :
 - située à la date du début du run,
 - ayant la valeur de la première donnée.
 - Si la date de la dernière donnée est inférieure à la date de fin du run, ajouter une donnée :
 - située à la date de fin du run,
 - ayant la valeur de la dernière donnée.
- ✓ Si l'option « init_rampe » est activée (cf §3.6.1) :
 - ↳ Si le code entité est listé dans la clé « stations » de la clé « rampe-sections » trouvée :
 - Ajouter une donnée au temps défini, avec la valeur définie :
 - le temps correspond à la clé « time » du « rampe-sections » ; la valeur du temps est modifiée pour être négative.
 - La valeur correspond à la valeur associée à l'entité dans la clé « stations »

Exemple d'initialisation RAMPE avec les paramètres :

```
[init-rampe]
rampe-sections= rampe1

[rampe1]
file = LigneEauInitiale_1.lig
time = 86400
stations = H5172010;60|H5091010;10|H520100501;3.6
```

Lors du calcul avec le fichier « LigneEauInitiale_1.lig » correspondant à « rampe1 », les 3 fichiers obtenus sont :

H5172010.loi :

```
# H5172010
# Temps (s) Q
      S
-86400  60.0
0       [debit-H5172010-1]
```

```
3600    [debit-H5172010-2]
...
```

H5091010.loi :

```
# H5091010
# Temps (s) Q
      S
-86400 10.0
0      [debit-H5091010-1]
3600   [debit-H5091010-2]
...
```

H520100501.loi :

```
# H520100501
# Temps (s) Z
      S
-86400 3.6
0      [hauteur-H520100501-1]
3600   [hauteur-H520100501-2]
...
```

3.6.4.3 Modification du fichier « ParametresMascaret.xcas »

Le fichier « ParametresMascaret.xcas » est généré dans le « **répertoire de travail PIM** » par copie et modification du fichier « ParametresMascaret.xcas » du répertoire de configuration du modèle Mascaret (répertoire « config »).

Ce fichier contient des références relatives vers des fichiers ou répertoires de paramétrage. Le lancement par le PIM respecte l'arborescence standard du paramétrage Mascaret, ce qui permet d'éviter d'avoir à mettre à jour ces informations.

Sur la base de cette copie, certaines valeurs sont mises à jour :

- ✓ Balise « **fichierCas** → **parametresCas** → **parametresGeneraux** »
 - ↳ Sous balise « code » : détermine le noyau de calcul à utiliser. Cette valeur est inchangée, sauf pour le régime permanent où elle devient « 1 » (cf. 1.9.2)

Note : le fichier référencé dans la balise « fichMotsCles » est le fichier en cours de lecture. Il n'est pas à modifier.

- ✓ Balise « **fichierCas** → **parametresCas** → **parametresTemporels** »

Si l'initialisation RAMPE est active (cf §3.6.1) :

- Calculer la variable DATE_DEBUT_RAMPE = valeur négative de l'option « time » de la « rampe-sections » en cours
- Sous balise « **tempsInit** » : DATE_DEBUT_RAMPE
- Sous balise « **tempsMax** » (équivalent de DATE_FIN_SIMULATION) : nombre de secondes de différence entre DATE_FIN_SIMULATION et DATE_DEBUT_RAMPE.
- Sous balise « **nbPasTemps** » : calculé en faisant (DATE_FIN_SIMULATION - DATE_DEBUT_RAMPE) / PDT

Si l'initialisation RAMPE n'est PAS active (cf §3.6.1) :

- Sous balise « **tempsInit** » (temps relatif de DATE_DEBUT_SIMULATION) : 0
 - Sous balise « **tempsMax** » (équivalent de DATE_FIN_SIMULATION) : nombre de secondes de différence entre DATE_FIN_SIMULATION et DATE_DEBUT_SIMULATION.
 - Sous balise « **nbPasTemps** » : calculé en faisant DATE_FIN_SIMULATION / PDT (le temps DATE_DEBUT_SIMULATION étant pris égal à 0)
- ✓ Balise « **fichierCas** → **parametresCas** → **parametresCasier** », si la balise « **presenceCasiers** » est « true »
- ↳ Sous balise « **fichierGeomCasiers** » : le nom du fichier des casiers. Théoriquement, ce nom de fichier est inchangé (« GeometrieCasiers.casier »)
 - ↳ Sous balise « **cotesInitiale** » : la liste des cotes initiales des casiers ; 1 valeur par casier, dans l'ordre de numérotation des casiers.
 - En mode DEF AUT : ne pas modifier cette balise.
 - En mode REPRIS E : mettre le contenu du fichier « .casiers_pim » associé à la ligne d'eau trouvée (cf § 3.6.5). Si ce fichier n'est pas présent, cela génère une erreur pour cette méthode d'init et passe à la méthode d'init suivante.
 - En mode Prevision : mettre le contenu du fichier « .casiers_pim » correspondant au calcul d'analyse. Si ce fichier n'est pas présent, cela génère une erreur bloquante pour ce calcul.
- ✓ Balise « **fichierCas** → **parametresCas** → **parametresConditionsInitiales** »
- ↳ Sous balise « **ligneEau** → **LigEaulnit** » : « true » en cas de reprise sur une ligne d'eau, « false » sinon (mode permanent uniquement)
 - ↳ Sous balises de « **ligneEau** → **fichLigEau** » : « LigneEaulnitiale_in.lig »
- ✓ Balise « **fichierCas** → **parametresCas** → **parametresSingularite** »
- ↳ Dans le cas d'un run spécial barrage
 - ↳ Dans le cas d'un run de prévision ou d'un run d'analyse en mode « REPRIS E »
 - ↳ Pour chaque barrage issu du fichier d'archive des cotes de crêtes « .barrages_pim » adéquat, mise à jour de la valeur de la balise « **seuils** → **structureParametresSeuil** » → « **coteCrete** »

3.6.4.4 Affichage dans le fichier « pom.log »

- ✓ Pour un calcul SANS casiers
- ↳ Indiquer le nom du fichier d'archive utilisé :
« Reprise du fichier ' xxxx.pim ' »
- ✓ Pour un calcul AVEC casiers
- ↳ En mode DEF AUT : indiquer
« Cotes initiales de tous les casiers à T0 par défaut : » + contenu de la balise « **cotesInitiale** » au format « CAS 1=X ; CAS 2=X ; ... ».
 - ↳ En mode REPRIS E : indiquer

« Reprise du fichier 'xxxx.pim'. Cotes initiales de tous les casiers à T0 par défaut : » + contenu du fichier « .casiers_pim » utilisé au format « CAS 1=X ; CAS 2=X ; ... ».

↳ En calcul prévision : indiquer

« Cotes initiales à T1 : » + contenu du fichier « .casiers_pim » utilisé au format « CAS 1=X ; CAS 2=X ; ... ».

3.6.5 Run d'analyse : choix du fichier ligne d'eau

La préparation des fichiers « .loi », « .xcas » est faite par le RunconfigPimProcessor.

Dans le fichier « ParametresMascaret.xcas », le fichier de ligne d'eau a toujours le même nom « LigneEauInitiale_in.lig ».

Si le RunconfigPimProcessor est configuré avec un fichier de ligne d'eau qui existe (en mode « reprise », ou mode « défaut »),

- ✓ il va copier ce fichier ligne d'eau vers le fichier « {répertoire de travail Mascaret}/LigneEauInitiale_in.lig » puis lancer le calcul Mascaret.

Sinon, (le RunconfigPimProcessor n'a pas de fichier de ligne d'eau ou un fichier de ligne d'eau inexistant) :

- ✓ le RunconfigPimProcessor va avertir l'utilisateur avec le message dans le fichier « pom.log » :

« Le calcul d'analyse avec le fichier ligne d'eau '{FICHIER}' n'est pas possible car ce fichier ligne d'eau n'existe pas ».

- ✓ le RunConfigPimProcessor va arrêter ce calcul et laisser continuer la suite du traitement général.

3.6.6 Lancement du run

En pré-requis du lancement d'un run, les variables d'environnement « PYTHONPATH » et « LD_LIBRARY_PATH » doivent être positionnées pour l'utilisateur système d'exécution « mascaret ». Ceci est réalisé sur la VM Mascaret (cf MIEX), ce n'est pas à faire dans la commande définie sur la plateforme dans la POM.

Les étapes réalisées par le PIM pour le lancement d'un run par Mascaret sont :

- ✓ Génération du script « calcul.py » à partir d'un script modèle, avec les valeurs spécifiques pour le calcul :

↳ le chemin complet du dossier de calcul

↳ la liste des fichiers d'entrée ; cela comprend les types :

- **xcas** : le fichier xcas courant
- **geo** : depuis la balise « parametresGeometrieReseau/geometrie/reseau »

- **loi** : depuis les balises
« **parametresLoisHydrauliques/lois/structureParametresLoi/** » ; l'ordre de déclaration dans le fichier « .xcas » doit être gardé.
- **listing** : d'après la balise « **parametresImpressionResultats/listing** »
- **res** : d'après la balise « **parametresImpressionResultats/resultats** »

si la balise « **parametresGeneraux/presenceCasiers** » est à « true » :

- **casier** : d'après la balise « **parametresCasier/fichierGeomCasiers** »
- **listing_casier** : d'après la balise
« **parametresImpressionResultats/casier/listingCasier** »
- **listing_liaison** : d'après la balise
« **parametresImpressionResultats/casier/listingLiaison** »
- **res_casier** : d'après la balise
« **parametresImpressionResultats/casier/resultatCasier** »
- **res_liaison** : d'après la balise
« **parametresImpressionResultats/casier/resultatLiaison** »

↳ la définition du fichier ligne d'eau :

- « **LigneEaulInitiale_in.lig** » s'il y a un fichier ligne d'eau
- « » si aucun fichier ligne d'eau (cela correspond au mode 1_SARAP) ; dans ce cas, le script « calcul.py » n'effectue pas l'appel à
« **init_hydro_from_file()** », mais effectue une initialisation par défaut du modèle Mascaret.

↳ les dates nécessaires au calcul : la date de début simulation, la date de fin de simulation, le pas de temps, qui correspondent aux valeurs des balises « **tempsInit** », « **tempsMax** », « **pasTemps** » du fichier « .xcas »

✓ Copie du dossier « Structures » à côté de « calcul.py ». Ce dossier est fourni par Artelia et contient du code Python utilisé par « calcul.py ».

✓ Exécution du script « calcul.py ».

↳ Ce script utilise l'API Python pour lancer Mascaret.

3.6.7 Lancement du run spécial barrages

Le PIM ne lance le run spécial barrages que si le fichier nommé « **Fichier_Barrage_Mobile.txt** » est présent dans le dossier modèle mascaret.

Les étapes réalisées par le PIM pour le lancement d'un run spécial barrage sont :

- ✓ vérification de l'option « **cheminmascaretbarrage** » : si elle ne correspond pas à un fichier, le PIM génère une erreur bloquante
- ✓ copie de l'exécutable mascaret spécial barrages, défini par l'option « **cheminmascaretbarrage** », vers le dossier de calcul.

✓ Génération du script « calcul.py » à partir d'un script modèle, avec les valeurs spécifiques pour le calcul :

↳ le chemin complet du dossier de calcul

- ✓ Exécution du script « calcul.py ».
- ↳ Le script « calcul.py » n'utilise pas les API Python de mascaret.
- ↳ Il génère le fichier relais à partir du fichier résultat

Le run spécial barrages ne gère pas :

- ✓ l'init rampe : le PIM génère une erreur bloquante dans ce cas
- ✓ la génération des fichiers ligne d'eau hors fin de calcul : le PIM affiche un warning dans les logs

3.6.8 Vérification de la bonne exécution du run

Le script « calcul.py » (standard ou spécial barrage) intercepte les éventuelles erreurs de calcul dans Mascaret et écrit sur stdout :

- ➔ soit une ligne avec « FIN » pour indiquer un calcul sans erreur
- ➔ soit une ligne «ERREUR : » suivi du message d'erreur récupéré de Mascaret.

Le PIM analyse la sortie standard stdout pour déterminer si le calcul s'est bien déroulé ou si un erreur est à remonter dans la POM.

3.7 Traitement des sorties

Cette étape vise à produire les fichiers de sortie attendus par la POM à partir des résultats générés par Mascaret.

- ✓ Résultats numériques à intégrer au(x) fichier(s) « .xml » attendu(s) par la POM
- ✓ Variables d'assimilation
- ✓ Archive ZIP éventuellement
- ✓ Archivage de la ligne d'eau

3.7.1 Sorties de type « fichier »

Le PIM vérifie que le fichier « parameters.xml » contient au plus une sortie de type « fichier ». Dans le cas contraire une erreur est levée et le calcul s'arrête.

Si une sortie est de type fichier, le PIM ne génère aucun fichier zip.

Une avertissement est écrit dans le progression.xml.

3.7.2 Sorties XML Sandre

Pour chaque run analyse/prévision, les résultats numériques sont produits dans le fichier:

Resultats/ResultatsOpthyca.opt

Le nom « ResultatOpthyca.opt » est prédéfini dans chaque modèle mascaret ; le PIM le définit dans le fichier .ini avec la clé 'resultfilepalm'.

- ✓ Ce fichier est au format « opt » (cf. 1.6.3.2). Il contient des résultats pour les stations paramétrées dans le fichier « .ini » dans le paramètre « stationsresultat » (cf. 1.6.2)

Ce fichier est lu par les méthodes `pix_get_hydrodata_analyse()` et `pix_get_hydrodata_prevision()` définies par le PI 3 et surchargées par le PIM.

Pour un run de prévision, les résultats concernent la série propagée par ce run.

Les valeurs produites par Mascaret sont en « m » pour les hauteurs, en « m³/s » pour les débits (valeurs à convertir car le XML sandre contient des hauteurs en « mm » et des débits en « l/s »).

3.7.2.1 Décalage vertical sur les prévisions

Le PIM permet d'effectuer une conversion d'échelle sur les hauteurs prévues.

Pour cela, dans le paramètre **stationsresultat**, pour chaque tuple, la troisième valeur correspond au décalage à appliquer en m : « code station; code section ; décalage vertical en m »

Le paramètre décalage vertical n'a pas de sens pour les sites. Un message d'avertissement dans « progression.xml » et dans « pom.log » signale l'incohérence : « Pour le site xxxx, la conversion d'échelle ne peut pas être appliquée sur les prévisions ». Ce message n'est pas bloquant, les calculs se poursuivent.

Le PIM ne contrôle pas que pour une station donnée, il y a bien un décalage sur les entrées, paramètre **stationsloi**, et sur les prévisions, paramètre **stationsresultat**.

Pour chaque décalage qui est appliqué sur une station et une hauteur, un message d'information dans « progression.xml » et dans « pom.log » est tracé : « Pour la station xxx, la conversion d'échelle est appliquée sur les prévisions avec le décalage 'valeur' m », un seul message pour toute la série (ie pas de message pour chaque date)

Pour chaque prévision de la station, le décalage en m est ajouté à la prévision.

3.7.3 Archivage de la ligne d'eau

Avec le paramétrage défini par le PIM, le script « calcul.py » va créer un ou plusieurs fichiers ligne d'eau lors d'un calcul d'analyse.

Le paramètre « backuptimesteps » permet de définir la génération de plusieurs fichiers ligne d'eau. Chaque fichier généré est nommé « relais_AAAAMMDD_HHMM.lig », la date correspond à la date de cet état, c'est à dire la date de fin de simulation.

La génération de ces fichiers est effectué dans le script « calcul.py » par des lancements successifs de Mascaret et pour chaque lancement par la génération d'un fichier ligne d'eau.

Pour chaque lancement de Mascaret, les dates de début de fin de simulation sont définies comme suit :

```
backuptimesteps=-valeur_1|.....|-valeur_n|0
```

Lancement	Date de début	Date de fin
1 ^{er}	T0	T1 - valeur_1
i ^{ème}	T1 - valeur_i-1	T1 - valeur_i
dernier	T1 - valeur_n	T1

Tableau 4 : archivages des lignes d'eau

Dans le cas où plusieurs fichiers lignes d'eau sont générés, un message signale à l'utilisateur la génération de ces fichiers dans « progression.xml » :

« Le fichier ligne d'eau « nom_du_fichier.lig » a été généré durant le run d'analyse »
avec un message pour chaque fichier dès lors qu'il y en a plus que un.

Si le calcul Analyse est terminé sans erreur, le PIM récupère les fichiers ligne d'eau créés et va les archiver dans le dossier « archives », comme indiqué au chapitre 1.7.2 Arborescence de travail du PIM :

- ✓ les fichiers sont regroupés dans un dossier pour chaque code modèle Mascaret :
« archives/{CODE_MODEL_MASCARET} »
- ✓ chaque nom de fichier ligne d'eau est nommé :
{TPS_DATE_FIN}__{TYPE_SESSION}__{SCENAR_POM}__{MODE}.pim : le PIM renomme chaque fichier nommé « relai_AAAAMMDD_HHMM.lig » en
« AAAAMMDD_HHMM__{TYPE_SESSION}__{SCENAR_POM}__{MODE}.pim ».
- ✓ Si un fichier de même nom existe déjà dans le dossier
« archives/{CODE_MODEL_MASCARET} », le PIM le remplace par le nouveau.

Si le calcul est terminé sans erreur et s'il y a un fichier « .cas_opt » (d'après la balise « parametresImpressionResultats/casier/resultatCasier »), le PIM va générer des fichiers « .casiers_pim » dans le dossier « archives », comme indiqué au chapitre 1.7.2 Arborescence de travail du PIM :

- ✓ les fichiers sont regroupés dans un dossier pour chaque code modèle Mascaret :
« archives/{CODE_MODEL_MASCARET} »
- ✓ le PIM lit le fichier « .cas_opt » (cf format § 1.6.3.3)
- ✓ Pour chaque élément du paramètre « backuptimesteps », le PIM cherche dans le fichier « cas_opt » les données des casiers correspondant à ce temps. S'il en trouve, le PIM va écrire les données de la colonne « ZCAS » dans un fichier
« AAAAMMDD_HHMM__{TYPE_SESSION}__{SCENAR_POM}__{MODE}.casiers_pim » (cf format § 1.6.3.4).
- ✓ Si un fichier « .casiers_pim » de même nom existe déjà dans le dossier
« archives/{CODE_MODEL_MASCARET} », le PIM le remplace par le nouveau.

3.8 Finalisation

Cette étape est identique au PI. Il s'agit de la mise à jour du fichier de progression de manière à signifier à la POM la fin de l'exécution du modèle : statut à 0, avancement à 100% (avec éventuellement un message de fin d'exécution).