



# **Hélios**

## **Présentation des schémas XML du PES**

**Septembre 2005**



## Table des Matières

<b>1. Objet.....</b>	<b>4</b>
<b>2. Introduction.....</b>	<b>8</b>
2.1 Contexte général.....	8
2.2 L'échange des données dans Hélios.....	8
<b>3. Les objectifs du PES.....</b>	<b>8</b>
<b>4. La démarche Hélios.....</b>	<b>9</b>
<b>5. Les concepts de base de la modélisation.....</b>	<b>9</b>
5.1 La notion d'objet dans Hélios.....	9
5.2 Le sens des échanges.....	10
5.3 L'unité d'échange dans Hélios : le message.....	10
5.4 La cinématique d'échange.....	10
5.5 La sécurité des échanges.....	11
5.6 Les canaux de communication.....	11
5.7 La gestion des versions du PES par Hélios.....	11
5.8 Le passage au PES.....	11
<b>6. présentation de l'architecture logique du serveur d'échange.....</b>	<b>12</b>
<b>7. Principes &amp; normes retenus par Hélios.....</b>	<b>13</b>
7.1 Le choix de XML-Schéma.....	13
7.2 Le numéro de version XML.....	13
7.3 Le Jeu de caractères (« encoding »).....	13
7.4 Les espaces de noms dans Helios.....	14
7.5 Le choix attributs vs éléments.....	14
7.6 La représentation des données élémentaires.....	14
7.7 Le nommage des éléments et attributs XML.....	15
7.8 Les caractères réservés.....	15
7.9 La transformation des classes UML en XML.....	16
<b>8. Techniques de construction des schémas XML du PES.....</b>	<b>16</b>
8.1 Introduction.....	16
8.2 La construction des composants réutilisables.....	16
8.3 L'assemblage par domaine fonctionnel.....	17
8.4 La construction du répertoire des messages PES.....	18
<b>9. Les domaines et les objets couverts.....</b>	<b>20</b>
9.1 Les domaines et les objets métier modélisés.....	20
9.2 Les domaines et les objets transverses modélisés.....	22



<b>10. La gestion des versions de la modélisation pour l'ADAE.....</b>	<b>22</b>
<b>11. Les coordonnées pour contacter Hélios.....</b>	<b>22</b>
<b>12. Annexe A : .....</b>	<b>23</b>
<b>13. Annexe B .....</b>	<b>23</b>

## 1. Objet.

---

Ce document sert à la publication de la première version définitive du PES et retrace les travaux de modélisation XML menés dans le cadre du programme Hélios de la Direction Générale de la Comptabilité Publique (DGCP).

Cette publication des schémas XML intègre les remarques apportées par les éditeurs sur la version précédente et les derniers enrichissements liés aux évolutions réglementaires applicables au 1<sup>er</sup> janvier 2006 dans le cadre de la réforme des instructions M14, M52 et M61. Ces dernières ont été publiées par ordonnance du 27 août 2005 et visent à la simplification et à l'amélioration des règles budgétaires et comptables applicables aux collectivités territoriales, à leurs groupements et aux établissements publics locaux qui leur sont rattachés. Les mesures prises concernent notamment la simplification des opérations d'ordre, des règles de provisionnement et la possibilité de reprendre un excédent d'investissement en section de fonctionnement.

Cette publication doit permettre de communiquer aux partenaires informatiques du secteur local public la première version définitive du PES amendée des dernières mises à jour.

Les modifications apportées aux schémas XML et aux codifications à utiliser sont récapitulées dans les tableaux ci-dessous dont le premier isole les incompatibilités avec la version précédente des schémas.

Les tableaux font apparaître :

- La référence de l'évolution dans le dossier PES
- Son identification
- Un commentaire éventuel

**Tableau des évolutions avec un impact sur la compatibilité de la version 1 du PES**

Référence	Identification	Commentaires
Paragraphe 4.2.3	Schéma XML – Class bancaire_1.1	Le type numéro de compte est Base_Texte pour permettre l’insertion de caractères « espaces ».
Ensemble du document	Ensemble du document	Le type Affect est en base texte pour permettre l’insertion de caractères « espaces »
Paragraphe 4.3.6.2 et 4.4.6.2	Schémas XML : <ul style="list-style-type: none"> <li>- Class_Ligne_Mandat_Aller</li> <li>- Class_Ligne_Mandat_Retour</li> <li>- Class_Ligne_Titre_Aller</li> <li>- Class_Ligne_Titre_Retour</li> </ul>	Les types IdRegie, IdConv et IdEmpruntOrdo sont en Base_Texte pour permettre l’insertion de caractères « espaces » La longueur maximale du champ IdEmpruntOrdo est étendue à 24 caractères.

Paragraphe 4.3.8, 4.4.8, 4.5.8, 4.8.5 et 4.9.4.2	Schémas XML : <ul style="list-style-type: none"> <li>● Class_TiersDepense.xsd</li> <li>● Class_TiersRecette.xsd</li> <li>● Class_TiersRole.xsd</li> <li>● Class_TiersMarche.xsd</li> <li>● Class_Ecriture.xsd</li> </ul>	Le type RefTiers est Base_Texte pour permettre l’insertion de caractères espaces . La longueur maximale est homogénéisée à 30 caractères.
Paragraphe 4.3.2.1 et 4.4.2.1	Schémas XML : <ul style="list-style-type: none"> <li>● Class_LigneTitreAller</li> <li>● Class_LigneTitreRetour</li> <li>● Class_LigneMandatAller</li> <li>● Class_LigneMandatRetour</li> </ul>	Evolutions réglementaires au 1 <sup>er</sup> janvier 2006 des instructions M14, M52 et M61. Des valeurs sont réservées dans les zones opérations des lignes de pièce des domaines « recette » et « dépense » afin d’identifier les chapitres de contrôle.
Paragraphe 4.3.2.1 et 4.4.2.1	Codification de la zone nature « NatPce » pièce des domaines « recette » et « dépense »	Evolutions réglementaires au 1 <sup>er</sup> janvier 2006 des instructions M14, M52 et M61. Ajout d’une codification 18 – opération d’ordre liée aux cessions.

Paragraphe 4.6.5	Restructuration du schéma XML Class_Ligne_Budget_1.1. en deux : <ul style="list-style-type: none"> <li>- Class_Ligne_Budget_Aller et</li> <li>- Class_Ligne_Budget_Retour</li> </ul>	Objectifs : le montant réalisé est une information retour exclusivement La zone Nature devient facultative Les types LibCpte et ContNat sont en Base_Texte pour permettre insertion de caractères espaces
------------------	---	--

### **Tableau des autres modifications**

Paragraphe 4.3.2	Schéma XML – CommunDepense_1.1 Ajout d'une nouvelle codification « 11 : virement interne » comme Mode de règlement (balise ModRegl)	Une telle codification est indispensable pour le traitement des mandats de paie.
Paragraphe 4.4.2 et 4.5.2	Codification des domaines recette et role.	Des ajouts ont été réalisés dans les codifications des zones Code mise en instance et rejet « CodInst-Rej » et Code suspension de poursuite « CodSuspPour ». <ul style="list-style-type: none"> <li>● Nouvelle codification de la zone « CodInst-Rej »</li> <li>● Nouvelle codification de la zone « CodSuspPour »</li> </ul>
Paragraphe 4.7.1 à 4.7.4	<ul style="list-style-type: none"> <li>● La zone 'CatImmo' (catégorie immobilisation) devient une codification libre et passe de 2 caractères à 6 caractères alphanumériques (zone saisie libre).</li> <li>● Conséquence sur schémas XML CommunInventaire et ClassInventaire</li> <li>● Class Inventaire : le type NumInvent est Base_Texte pour permettre l'insertion de caractères « espaces ».</li> </ul>	
Paragraphe 4.7.1 à 4.7.4	<ul style="list-style-type: none"> <li>● La durée 'DurAmt' (durée de l'amortissement ) n'est plus exprimée en mois mais en années (3 caractères)</li> </ul>	

Paragraphe 4.8.2	Class CommunMarche : la codification TypDep (catégorie de marché) est refondue pour permettre la distinction entre Travaux Opérations et Travaux Ouvrage	
Paragraphe 4.10.4	Corrections dans le schéma XML ClassEmprunt_1.1.xsd <ul style="list-style-type: none"><li>• Suppression du champ EmpDev</li><li>• Insertion de balise Interet dans le bloc LigneTableauAmt</li><li>• Correction du typage des données FraisEch, Capital, MtCrd</li><li>• Les types NumEmpOrdo et NumEmpPret sont Base_Texte pour permettre l'insertion de caractères espaces. Leur longueur maximale est étendue à 24 caractères.</li></ul>	

Cette publication est structurée comme suit :

Un document de présentation récapitulatif :

- La démarche de mise en œuvre d'XML pour Hélios.
- Les concepts utilisés pour la modélisation.
- Les normes appliquées.
- La technique de construction et le périmètre fonctionnel couvert.

Des documents annexes détaillant le volet technique des travaux et l'implémentation réalisée. Ce volet technique est illustré avec des exemples.

## **2. Introduction.**

---

### **2.1 Contexte général.**

Au sein du MINEFI, la Direction générale de la comptabilité publique (DGCP) et les services déconcentrés qui lui sont rattachés forment le réseau du Trésor Public. Ce réseau assure des missions diversifiées portant sur les opérations de l'Etat et les opérations du Secteur Public Local, collectivités locales et établissements publics locaux (EPL).

Les applications existantes pour le secteur local se sont construites par strates successives et ont atteint leurs limites fonctionnelles et techniques. Elles ne répondent plus aux besoins et aux exigences de service et de qualité mise en œuvre au Trésor.

La refonte du système d'information dédié au secteur local menée à travers le programme Hélios est l'occasion pour refondre tous les protocoles des applications à remplacer en un seul et unique protocole dénommé « Protocole d'échange standard » (PES).

### **2.2 L'échange des données dans Hélios.**

Hélios est une application communicante. Le futur protocole d'échanges d'informations avec les partenaires se caractérise par :

- Son « unicité » dans la couverture de l'ensemble des besoins repartis aujourd'hui dans de multiples protocoles et applications,
- Une plus grande richesse des informations échangées,
- Un partage et une évolutivité plus aisée à travers l'adoption du format XML.

## **3. Les objectifs du PES.**

---

Le protocole PES sera le nouveau format pivot des échanges entre Hélios et ses partenaires (internes et externes à la DGCP) au premier rang desquels se trouvent les ordonnateurs. Le PES :

- Est un véhicule de communication dans les deux sens entre Hélios et ses partenaires.
- Est la synthèse de l'ensemble des interfaces supportées par les applications à remplacer.

- Intègre les nouvelles données suite aux évolutions juridiques et réglementaires concernant le secteur local.
- Intègre les données nécessaires à la mise en œuvre des nouveaux services apportés par Hélios.
- Remplace à l'issue d'une phase transitoire (2006) les protocoles existants et devient le seul protocole d'échange.
- Couvre l'ensemble des fonctionnalités offertes par Hélios à savoir :

Fonctionnalités de base	Nouvelles fonctionnalités
Budget	Immobilisations
Comptabilité	Emprunts
Dépense	Régies
Recette	Marchés
Recouvrement	

Cette publication intègre les domaines et objets du périmètre PES version 1. Elle s'enrichira au fur et à mesure du développement des nouvelles versions.

## 4. La démarche Hélios.

---

La démarche développée par Hélios s'appuie sur les fondamentaux suivants :

- La définition commune des données métier du PES entre l'équipe Hélios et les représentants des ordonnateurs.
- La qualification des données qui permet de définir, les données « minimum obligatoire » et les données adaptables localement en fonction de la relation comptable-ordonnateur et de la spécificité du partenaire.
- Toutes les données manipulées par l'application et échangées avec des partenaires (externes ou internes à la DGCP) seront modélisées en XML.

Leur mise en œuvre s'est concrétisée par :

- La constitution de groupe de travail Hélios/type d'ordonnateur (les collectivités locales, les établissements publics de santé et les HLM) qui ont procédé à la définition des données métier du PES,
- La constitution d'un groupe de travail regroupant les informaticiens, d'Hélios et des ordonnateurs, qui a procédé à la définition du format XML du PES.
- La communication du bilan des travaux réalisés, aux partenaires et à leurs prestataires de service informatique.
- L'adhésion à la démarche de l'ADAE avec la publication du schéma XML du PES.

## 5. Les concepts de base de la modélisation.

---

### 5.1 La notion d'objet dans Hélios.

Les objets manipulés dans Hélios sont de deux natures :

- Des objets métier : appartenant à un domaine fonctionnel précis (ex : budget, titre, mandat, ...)
- Des objets complémentaires appartenant aux domaines transverses ou utilisés par plusieurs domaines fonctionnels d'Hélios (référentiels, etc...). Ces objets complémentaires seront désignés sous le nom « d'objets transverses ».

## 5.2 Le sens des échanges.

Les données dans Hélios ont vocation à circuler dans les deux sens entre les ordonnateurs et leurs comptables. Toutefois, par souci de simplification et de clarté, nous avons retenu les définitions suivantes :

- Le sens « aller » : définit un échange dont l'émetteur est l'ordonnateur et le récepteur est Hélios.
- Le sens « retour » : définit un échange dont l'émetteur est Hélios et le récepteur est l'ordonnateur.

## 5.3 L'unité d'échange dans Hélios : le message.

Le message constitue l'unité d'échange entre l'ordonnateur et le comptable et cela dans les deux sens. Un message est toujours composé de :

- Une enveloppe technique décrivant les paramètres du message, la carte de visite de l'émetteur et la carte de visite du destinataire,
- Le corps du message qui contient les structures qui décrivent les données applicatives.

## 5.4 La cinématique d'échange.

L'étude du PES a mis en évidence les éléments suivants, qui servent de point d'entrée à la modélisation XML :

- Les demandes de traitement de l'ordonnateur (ordres de paiement, de recette, ...) sont soumises via un protocole Aller. Le support du protocole Aller est un fichier qui contient un message composé d'une ou plusieurs structures. Un message PES est relatif à un couple 'Collectivité Budget' unique. Un message Aller est autoporteur, en ce sens qu'il ne contient pas de références à des messages précédemment reçus ou émis.
- La réception d'un fichier ordonnateur par HELIOS se traduit, après contrôles, par une intégration des données en base.
- Une intégration réussie des données dans HELIOS signifie que ces dernières sont disponibles pour le comptable. L'émetteur doit pouvoir en être informé par un message d'acquiescement (ack). Il n'y a pas de retour arrière possible après intégration d'un fichier dans Hélios. Les seules modifications possibles sont celles permises fonctionnellement par l'application.
- Un échec d'intégration des données dans HELIOS se traduit par une notification de non-acquit (nack) à l'émetteur (s'il est identifié). Il n'y a pas de possibilité offerte d'intégration partielle.

- Les retours d'informations effectués par le comptable à destination de l'ordonnateur sont soumis via un protocole Retour. Les caractéristiques de ce protocole sont identiques à celles du protocole Aller.

## **5.5 La sécurité des échanges.**

La présente version du PES ne met pas en œuvre des fonctionnalités de cryptage applicatif des données échangées. La sécurité des échanges repose sur celle offerte par les supports de transmission.

Les fichiers peuvent être compactés selon un protocole compatible avec les extracteurs Winzip ou pKZip.

## **5.6 Les canaux de communication.**

Les données à échanger entre ordonnateurs et HELIOS sont véhiculées sous forme de fichiers informatiques : ceux-ci font l'objet de négociations entre le comptable et l'ordonnateur. Les principes suivants sont retenus dans cette version :

- Unicité du canal d'acheminement par ordonnateur.
- Conservation du canal d'acheminement asynchrone actuel CFT.
- Communication par disquette via le poste comptable.
- Transmission synchrone, via un formulaire HTML, pour les ordonnateurs disposant d'un accès Extranet/IP sur HELIOS. Dans ce cadre, les flux « retour » se matérialisent sous forme de fichiers mis à disposition de l'ordonnateur pour téléchargement HTTP.

## **5.7 La gestion des versions du PES par Hélios.**

Le PES est un protocole qui devra vivre et évoluer. Afin d'anticiper les futures évolutions, les principes suivants sont retenus :

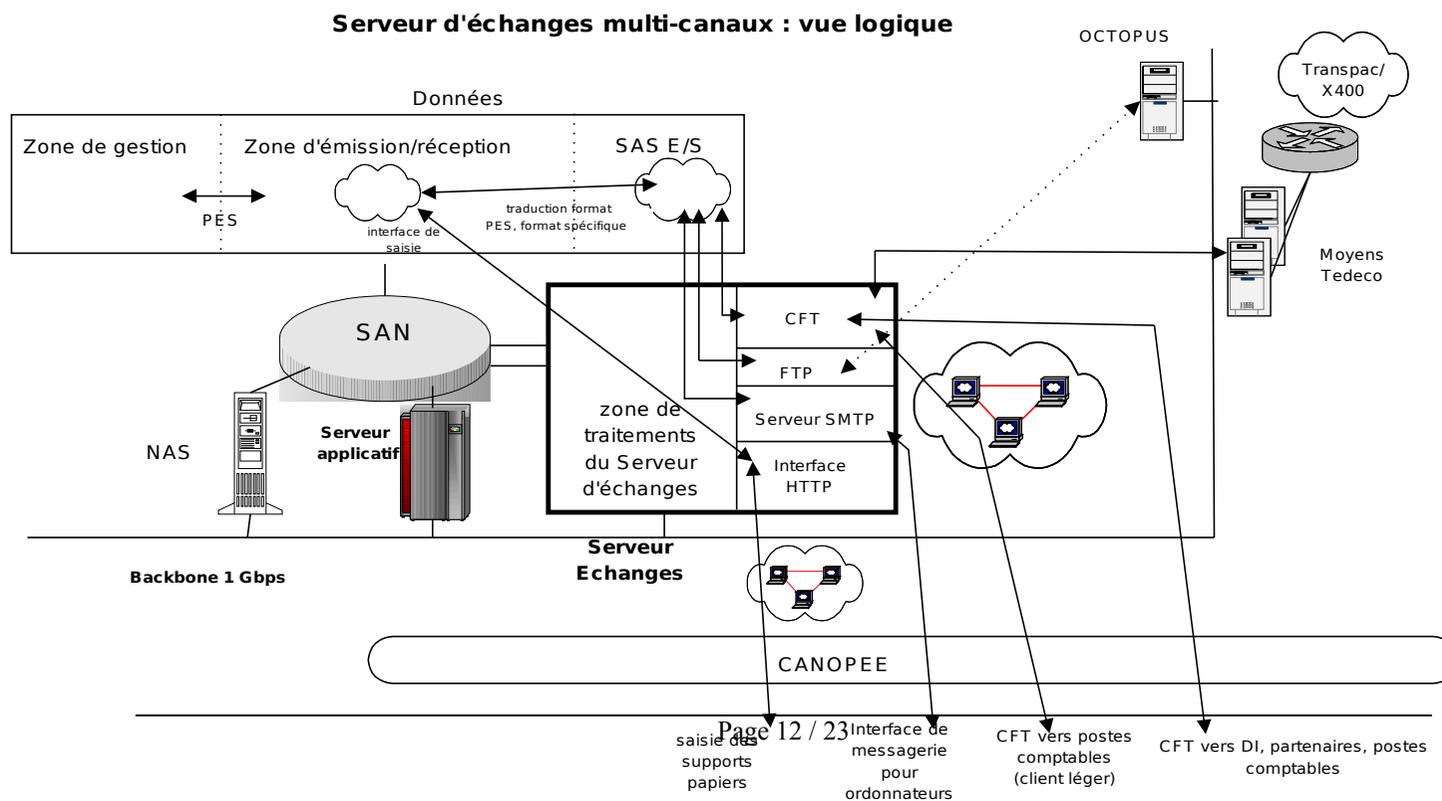
- Le PES est versionné par domaine fonctionnel. Une montée en version d'un domaine ne doit pas impacter les autres domaines si la nouvelle version ne concerne que des données de portée spécifique au domaine concerné.
- Pour un domaine fonctionnel donné, deux versions successives du PES peuvent exister simultanément.
- Pour un ordonnateur donné une, et une seule, version est utilisée à un instant donné.

## **5.8 Le passage au PES.**

Le passage au PES se fait pour les deux sens de la communication (aller et retour) simultanément.

## 6. présentation de l'architecture logique du serveur d'échange

Le schéma de l'architecture logique telle que définie pour supporter les échanges entre Hélios et ses partenaires.



## 7. Principes & normes retenus par Hélios.

---

### 7.1 Le choix de XML-Schéma.

Les messages du PES sont spécifiés en utilisant la notation XML Schéma. Plusieurs éléments ont conduit à retenir ce choix :

- L'objet du PES est de fournir un format pivot permettant l'échange de données et l'intégration de ces dernières dans HELIOS. Les données à échanger sont de différents types (décimaux, entier, booléen, code numérique ou alphanumérique, date, année,...) et doivent pouvoir faire l'objet d'enrichissement, de restriction sur des plages de valeurs. Un contrôle de typage des données est nécessaire avant introduction en base. La DTD offre des fonctionnalités très pauvres en matière de description de type (on ne va guère au-delà d'une chaîne textuelle). A l'inverse, XML Schéma est adapté à ce type de besoin : elle intègre 44 types de bases pré-construits, permet la définition de types complexes, la production de règles d'extension, de restriction, de réutilisation.
- La norme XML schéma, bien que récente, est stable, en tant que recommandation officielle du W3C, depuis mai-2001. Elle sera largement diffusée lors de la mise en œuvre du PES.
- La production XML réalisée dans le cadre du W3C ou dans le cadre des projets « Open Source » est basée sur l'utilisation de XML schéma.
- La norme XML Schéma est aujourd'hui prise en compte par la plupart des analyseurs syntaxiques (dont Xerces) et les éditeurs XML.
- La norme XML Schéma est totalement compatible avec les espaces de noms ('NameSpace') mis en œuvre dans la modélisation du PES pour définir des jeux de balises de portée domaine.

### 7.2 Le numéro de version XML.

L'attribut version indique la version du langage XML utilisée.

Règle : L'attribut version prend la valeur : 1.0.

### 7.3 Le Jeu de caractères (« encoding »).

Le jeu de caractères par défaut utilisé par les applications XML est le jeu de caractères UTF-8. Ce jeu de caractères n'est pas adapté à l'utilisation des caractères latins. Ainsi, le caractère 'é' est exprimé par la chaîne : `&#233;` ; ou bien caractère 'Φ' est exprimé par la chaîne : `&#934;`

Le système HELIOS est destiné à gérer des informations en langue française, c'est-à-dire comportant des lettres accentuées. Par conséquent, le jeu de caractères défini par la norme ISO-8859-1 est plus adapté. Il est alors possible d'écrire directement dans un document XML des caractères accentués.

L'utilisation du jeu ISO-8859-1 à la place de UTF-8 impose les contraintes suivantes :

- Insérer une indication XML indiquant le jeu de caractères utilisé :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

- Utiliser un parseur XML capable d'interpréter ce jeu de caractères.

Règle : L'attribut encoding prend la valeur : ISO-8859-1

## 7.4 Les espaces de noms dans Helios.

La mise en œuvre des espaces de noms vise d'une part à répondre aux objectifs de modélisation autonome des domaines et d'autre part à masquer la complexité inhérente à cette utilisation dans les documents produits et reçus par les ordonnateurs.

A cet effet, les dispositions suivantes sont prises :

- Tous les espaces de nom Hélios doivent commencer avec une racine sous la forme : « **www.minefi.gouv.fr/cp/helios/pes/** »
- Chaque domaine du PES correspond à une espace de nom (par exemple « http://www.minefi.gouv.fr/cp/helios/PES/depense » pour le domaine Dépense).
- La grammaire définissant le protocole Aller/Retour de chaque Domaine référence un espace de nom défini.

## 7.5 Le choix attributs vs éléments.

Le choix retenu est que tous les attributs du modèle, quels qu'ils soient, seront représentés par des éléments XML. Cette orientation est optimisée comme suit : la valeur d'une information n'est pas encadrée par une balise début et fin, mais encapsulée sous la forme d'un attribut XML, appelé V.

Cette représentation permet une économie substantielle de balises et constitue la forme définitive pour la représentation des attributs du modèle.

## 7.6 La représentation des données élémentaires.

Les données élémentaires représentent les structures atomiques à partir desquelles les structures plus complexes (classes, messages) sont construites.

Ces données élémentaires sont représentées sous forme d'éléments vides XML accompagnés d'un attribut de nom V dont le contenu est la valeur de la donnée.

Les types XML de données correspondant sont construits sous forme de types complexes XML dont le rôle est de déclarer l'attribut V. Cet attribut est lui-même un type simple XML construit à partir des types pré-construit XML auxquels peuvent s'appliquer des contraintes (facettes). Le tableau suivant en présente la mise en œuvre :

<i>Type de données</i>	<i>Définition</i>	<i>Type pré-construit XML</i>	<i>Facettes</i>	<i>Nom du type XML</i>
Numérique	Chaîne de nombre de taille maximum ='max' sans espace ni ponctuation	String	MaxLength, pattern	Base_Num'max'
Alphanumérique	Chaîne de nombre et de lettres de taille maximum ='max' sans espace ni ponctuation	String	MaxLength, pattern	Base_Alphanum'max'

Type de données	Définition	Type pré-construit XML	Facettes	Nom du type XML
Texte	Chaîne de lettres et de nombres de taille maximum 'max' avec espace et ponctuation	String	maxLength	Base_Texte'max'
Décimal (15)	Montant sur 15 chiffres maximum dont 2 chiffres après la virgule	Decimal	TotalDigits, fractionDigits	Base_Montant15
Décimal (5)	Taux exprimé sur 5 chiffres maximum dont 2 chiffres après la virgule	Decimal	TotalDigits, fractionDigits	Base_Taux
Date	Date sous le format AAAA/MM/JJ	Date		Base_Date
Année	Année sous le format AAAA	GYear		Base_Année
Booléen	Choix de deux valeurs possibles (0 ou 1)	Boolean		Base_Boolean

L'ensemble des types déclarés sont regroupés dans un même schéma XML 'simple\_type.xsd'. Ils doivent être inclus (directive 'xsd:include') dans les classes utilisant ces données.

## 7.7 Le nommage des éléments et attributs XML.

La règle de désignation des noms d'éléments et/ou attributs XML, à partir des noms issus de l'activité de modélisation consiste à concaténer les différents mots en mettant en majuscule la première lettre de chaque mot et en minuscule les autres lettres. L'objectif est de limiter la longueur des noms de manière à maximiser la charge utile dans un document XML. Une simulation effectuée par nos soins donne une charge utile comprise [10, 25]% du poids du document XML. La norme d'usage est de 1/3 (~33%).

## 7.8 Les caractères réservés.

Un certain nombre de caractères sont interdits dans les données contenues dans un élément XML. Afin de distinguer les balises et le contenu d'un élément, on remplace les caractères suivants par, au choix, l'entité prédéfinie correspondante, ou l'unicode décimal ou hexadécimal :

Caractère remplacé	Entité prédéfinie	Unicode décimal	Unicode hexa
<	&lt;	&#60;	&#x3c;
>	&gt;	&#62;	&#x3e;
&	&amp;	&#38;	&#x26;
'	&apos;	&#39;	&#x27;
"	&quot;	&#34;	&#x22;

Les caractères retour chariot (&#xd), nouvelle ligne (&#xa), tabulation (&#x9) et espace (&#x20) sont normalement préservés et transmis correctement à l'application par le parseur XML, hormis les combinaisons &#xd et &#xa transformées en un unique &#xa (pour éviter des problèmes liés aux plates-formes hétérogènes).

Les caractères tabulation (&#x9) sont interdits dans les données contenues dans les éléments. Les caractères retour chariot et retour ligne sont autorisés.

## 7.9 La transformation des classes UML en XML.

Les règles de transformation d'un diagramme de classe en schéma XML sont exprimées comme suit :

- Une classe UML est représentée comme un élément XML, contenant toutes les classes immédiatement dépendantes.
- Un attribut UML est représenté comme un élément XML vide ayant un attribut XML nommé « V ».

## 8. Techniques de construction des schémas XML du PES.

### 8.1 Introduction.

Nous présentons dans ce chapitre les techniques mises en œuvre pour l'implémentation XML du PES. Ces techniques portent sur les points suivants :

- Implémentation de composants réutilisables. Chaque composant donne lieu à l'élaboration d'un schéma XML et constitue une unité de livraison et de maintenance.
- Définition des règles d'assemblage des composants par domaine fonctionnel.
- Agrégation des différents domaines fonctionnels pour former un espace de publication des schémas décrivant les structures du PES.
- Mise en place d'une organisation des schémas (entrepôt) en vue de publication.

### 8.2 La construction des composants réutilisables.

Le PES se traduit par une forte diversité de structures, obéissant à des grammaires indépendantes mais reprenant des blocs communs. L'approche préconisée consiste à construire les différentes syntaxes par assemblage de blocs. Chaque bloc fait l'objet d'une description autonome sous forme de schémas XML. Une telle approche offre une grande souplesse quant à la réutilisation des blocs.

Un bloc réutilisable est constitué sous forme de schéma XML. Il ne déclare pas d'espace de nom pour faciliter la réutilisation. Le présent document introduit les types de blocs réutilisables suivants.

- Types de bases, utilisés pour les données élémentaires.
- Référentiels de codification, spécifiques par domaine.
- Classes XML pour la modélisation des différents blocs.

Les différents blocs sont placés dans un répertoire de dépôt, en vue de leur réutilisation.

#### 8.2.1 Les types de base.

Les types de base définissent une structure atomique de donnée. Le principe retenu est de masquer dans les schémas appelant le mode de construction des types de base utilisés.

Pour cela, la démarche mise en œuvre consiste à :

- Identifier les différents types de données élémentaires existant dans la description fonctionnelle du PES.

- Etablir la modélisation XML correspondante. Afin d'éviter une trop grande multiplicité de schémas, ces types seront regroupés dans un même schéma (« Simple\_Type\_1.1.xsd »).

Pour permettre l'utilisation d'un type de base par un schéma appelant, ce dernier doit effectuer l'instruction `<xs:include schemaLocation="Simple_Type_1.1.xsd"/>` avant la déclaration des éléments constitutifs du schéma.

### 8.2.2 Le référentiel de codification.

La modélisation fonctionnelle du PES met en évidence qu'un certain nombre de champs sont de type code numérique, c'est à dire des valeurs restreintes à un ensemble fini. Ces champs sont valorisés de façon contextuelle à un domaine. Le principe est de masquer dans les schémas appelants le mode de construction de ces données. Pour cela, la démarche mise en œuvre a consisté à :

- Identifier les différents codes numériques existant dans la description fonctionnelle du PES, domaine par domaine.
- Etablir la modélisation XML correspondante. Les types ainsi modélisés sont regroupés par domaine dans un même schéma (par exemple « CommunDepense\_1.1.xsd »).

### 8.2.3 La modélisation en Classe UML.

Chaque bloc fonctionnel (suite séquentielle d'attributs UML) donne lieu à la constitution d'un schéma XML (classe XML) ayant pour objet de déclarer un type de données dont la description correspond au bloc. La démarche mise en œuvre a consisté à :

- Identifier les blocs à partir de la description fonctionnelle du PES.
- Modéliser sous forme de schéma. Les classes sous le nom de Class\_{nomdelaclass}\_1.1.xsd.

Chaque classe XML incorpore (directive `xs:include`) le schéma 'Simple\_Type\_1.1.xsd' et le cas échéant le schéma 'Commun\_{NomduDomaine}\_1.1.xsd'.

## 8.3 L'assemblage par domaine fonctionnel

La modélisation XML du PES est effectuée par domaine fonctionnel. Les règles d'assemblage consistent à :

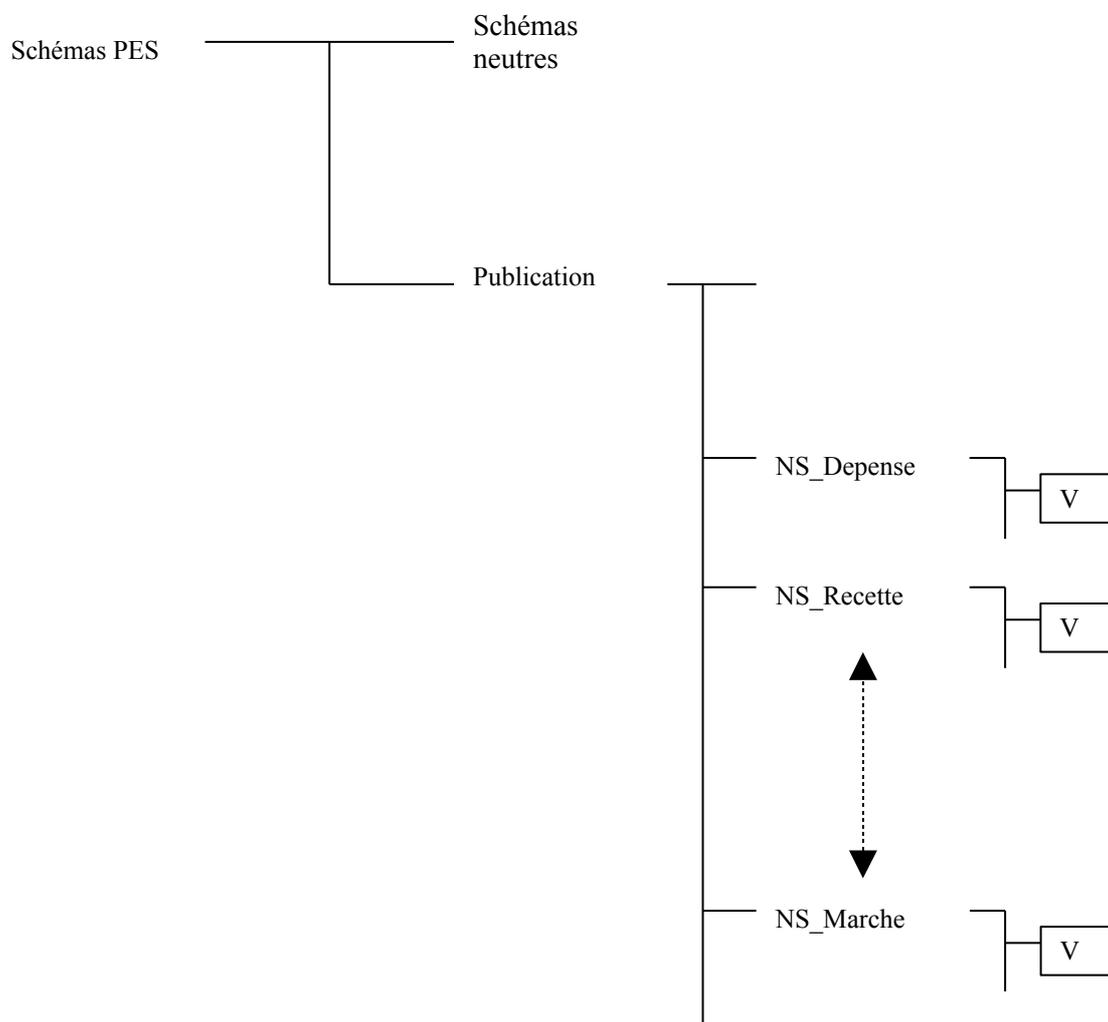
- Affecter un répertoire au domaine fonctionnel, qui contiendra les schémas utilisés par ce domaine.
- Effectuer un clonage (opération de « copier coller ») à partir du répertoire de dépôt des schémas nécessaires à la modélisation du domaine.
- Attribuer un espace de nom au domaine fonctionnel (par exemple « <http://www.minefi.gouv.fr/cp/helios/PES/Depense> »).
- Construire le schéma racine du domaine pour chacun des deux protocoles (Aller et Retour). Le nom de ce schéma est PES\_ 'NomduDomaine' Aller pour le protocole Aller (idem pour le retour). Ce schéma déclare un type de donnée correspondant à la portion de structure PES spécifique du domaine. Il déclare l'espace de nom du domaine et assemble les différents blocs nécessaires par « inclusion » (directive `xs:include`) dans le bloc d'en-tête du schéma.

## 8.4 La construction du répertoire des messages PES.

Les schémas XML décrivant les messages du PES (Aller, Retour, Ack & Nack) sont construits de la manière suivante :

- Chaque objet (métier ou transverse) identifié est décrit sous forme d'un schéma XML. Les schémas ainsi produits sont neutres en terme de 'NameSpace' (non déclaration d'espaces de noms cible, non déclaration d'espace de noms qualifiés par défaut) dans un objectif de réutilisation. La terminologie XML qualifie ces composants de « composants caméléons ». Le sous-répertoire « Schémas neutres » regroupe l'ensemble de ces composants.
- Chaque domaine fonctionnel du PES (Dépense, Recette, Rôle, Etat de l'actif, Etat du Passif, Budget, Marché, Comptabilité) regroupe un ensemble de composants issus, par clonage (« copier coller »), du répertoire « Schéma Neutres ». Ces composants sont regroupés dans un sous-répertoire de nom « {NS\_DF}/V{x}/{Ensemble de schémas}.xsd » avec :
  - 'NS\_DF' : NS\_{Nom du Domaine Fonctionnel} {Aller|Retour} (NS pour NameSpace).
  - V'x' : V{numéro de version pour le Domaine Fonctionnel}. Dans le cadre de la présente version du Protocole d'Echange Standard, il s'agit de la version '1'.
- Chaque domaine déclare un 'NameSpace' (désignation d'un ensemble de balises XML) correspondant au domaine fonctionnel.
- Chaque domaine réalise une inclusion (directive « xs:include ») des schémas constitutifs du domaine. L'ensemble des schémas ainsi inclus héritent du 'NameSpace ' du schéma racine.
- Chaque domaine positionne l'indicateur « ElementFormDefault » à « unqualified », de sorte que les instances d'éléments ne seront pas qualifiées (la référence aux NameSpace sera masquée dans les déclarations de balise) dans les documents instances, à l'exception de l'élément racine du document.
- Les schémas XML décrivant les structures du PES (PES Aller, PES Retour, Ack, Nack), sont stockés à la racine du sous-répertoire Publication. Ils incorporent (instruction « xs:import ») les schémas racines des différents domaines fonctionnels du PES. Les structures déclarent le 'NameSpace' « <http://www.minefi.gouv.fr/cp/helios/PES/> ».

### 8.4.1 Schéma de l'espace de mise à disposition.



## 9. Les domaines et les objets couverts.

### 9.1 Les domaines et les objets métier modélisés.

<i>Domaines métier couverts par Hélios</i>	<i>Objet modélisé dans le sens « ALLER »</i>	<i>Objet modélisé dans le sens « RETOUR »</i>
Budget	PES_BudgetAller_1.1.xsd (schéma de référence, possédant l'espace de nom du domaine) Class_BudgetAller_1.1.xsd Class_LigneBudgetAller_1.1.xsd Commun_Budget_1.1.xsd Simple_Type_1.1.xsd	PES_BudgetRetour_1.1.xsd (schéma de référence, possédant l'espace de nom du domaine Budget ) Class_BudgetRetour_1.1.xsd Commun_Budget_1.1.xsd Simple_Type_1.1.xsd
Comptabilité	Non défini	PES_Comptabilité_1.1.xsd (schéma de référence, possédant l'espace de nom du domaine Comptabilité ) Class_Compte_1.1.xsd Class_Ecriture_1.1.xsd Commun_Comptabilite_1.1.xsd Simple_Type_1.1.xsd
Dépenses	PES_DepenseAller_1.1.xsd (schéma de référence, possédant l'espace de nom du domaine Dépense ) Class_BordereauDepenseAller_1.1.xsd Class_MandatAller_1.1.xsd Class_LigneMandatAller_1.1.xsd Class_TiersDepense_1.1.xsd Class_Adresse_1.1.xsd Class_Bancaire_1.1.xsd CommunDepense_1.1.xsd Simple_Type_1.1.xsd	PES_DepenseRetour_1.1.xsd (schéma de référence, possédant l'espace de nom du domaine Dépense ) Class_BordereauDepenseRetour_1.1.xsd Class_MandatRetour_1.1.xsd Class_LigneMandatRetour_1.1.xsd Class_TiersDepense_1.1.xsd Class_Paiement_1.1.xsd Class_Adresse_1.1.xsd Class_Bancaire_1.1.xsd CommunDepense_1.1.xsd Simple_Type_1.1.xsd
Recettes	PES_RecetteAller_1.1.xsd (schéma	PES_RecetteRetour_1.1.xsd

<i>Domaines métier couverts par Hélios</i>	<i>Objet modélisé dans le sens « ALLER »</i>	<i>Objet modélisé dans le sens « RETOUR »</i>
	de référence, possédant l'espace de nom du domaine Recette) Class_BordereauRecetteAller_1.1.xsd Class_TitreAller_1.1.xsd Class_LigneTitreAller_1.1.xsd Class_RecouvrementRecetteAller_1.1.xsd Class_TiersRecette_1.1.xsd Class_Adresse_1.1.xsd Class_Bancaire_1.1.xsd Commun_Recette_1.1.xsd Simple_Type_1.1.xsd	(schéma de référence, possédant l'espace de nom du domaine Recette). Class_BordereauRecetteRetour_1.1.xsd Class_TitreRetour_1.1.xsd Class_LigneTitreRetour_1.1.xsd Class_ReceouvrementRecetteRetour_1.1.xsd Class_TiersRecette_1.1.xsd Class_Adresse_1.1.xsd Class_Bancaire_1.1.xsd Commun_Recette_1.1.xsd Simple_Type_1.1.xsd
Recouvrement	PES_RoleAller_1.1.xsd (schéma de référence, possédant l'espace de nom du domaine Role ) Class_Role_1.1.xsd Class_ArticleAller_1.1.xsd Class_SousArticleAller_1.1.xsd Class_RecouvrementRole_1.1.xsd Class_TiersRole_1.1.xsd Class_Adresse_1.1.xsd Class_Bancaire_1.1.xsd Commun_Role_1.1.xsd Simple_Type_1.1.xsd	PES_RoleRetour_1.1.xsd (schéma de référence, possédant l'espace de nom du domaine Role ) Class_Role_1.1.xsd Class_ArticleRetour_1.1.xsd Class_SousArticleRetour_1.1.xsd Class_RecouvrementRole_1.1.xsd Class_TiersRecette_1.1.xsd Class_Adresse_1.1.xsd Class_Bancaire_1.1.xsd Commun_Role_1.1.xsd Simple_Type_1.1.xsd
Immobilisations	PES_EtatActif_1.1.xsd (schéma de référence, possédant l'espace de nom du domaine Etat de l'Actif ) Class_Inventaire_1.1.xsd Commun_Inventaire_1.1.xsd Simple_Type_1.1.xs	Non défini
Emprunts	PES_EtatPassif_1.1.xsd (schéma de référence, possédant l'espace de	Non défini

<i>Domaines métier couverts par Hélios</i>	<i>Objet modélisé dans le sens « ALLER »</i>	<i>Objet modélisé dans le sens « RETOUR »</i>
	nom du domaine Etat du Passif ) Class_Emprunt_1.1.xsd Class_TiersEmprunt_1.1.xsd Class_Bancaire_1.1.xsd Commun_Emprunt_1.1.xsd Simple_Type_1.1.xsd	
Marchés	PES_Marche_1.1.xsd (schéma de référence, possédant l'espace de nom du domaine Marché ) Class_InformationMarche_1.1.xsd Class_TiersMarche_1.1.xsd Class_Adresse_1.1.xsd Class_Bancaire_1.1.xsd	Non défini

## 9.2 Les domaines et les objets transverses modélisés.

<i>Domaines transverses couverts par Hélios</i>	<i>Objet modélisé dans le sens « ALLER »</i>	<i>Objet modélisé dans le sens « RETOUR »</i>
EntetePES	Class_Enveloppe_1.1.xsd . Class_EnTetePES_1.1.xsd.	Class_Enveloppe_1.1.xsd Class_EnTetePES_1.1.xsd.
Adresse.	Classe_Adresse.xsd	Classe_Adresse.xsd
Compte bancaire	class_bancaire_1.1.xsd	Class_bancaire_1.1.xsd
Les type simples	Simple_Type_1.1.xsd	Simple_Type_1.1.xsd
Communication	Non défini	Class_Acquit_1.1.xsd Class_NonAcquit_1.1.xsd

## 10. La gestion des versions de la modélisation pour l'ADAE.

Dans la perspective des publications futures, nous proposons d'organiser l'arborescence de publication de l'ADAE à l'identique de l'arborescence de publication Hélios pour les ordonnateurs.

## 11. Les coordonnées pour contacter Hélios.



Suivant les recommandations de l'ADAE, Hélios met à disposition de tous les partenaires intéressés par ses travaux de modélisation une boîte à lettre générique pour adresser leurs contributions.

[Helios.xml@cp.finances.gouv.fr](mailto:Helios.xml@cp.finances.gouv.fr)

**Adresse postale de l'équipe Hélios :**

Ministère de l'économie des finances et de l'industrie  
DGCP Projet secteur local -HELIOS-  
Immeuble MAILLE NORD 3  
9-10, porte de Neuilly 93192 Noisy le Grand cedex  
Tél. : 01.58.84.77.23 (secrétariat) Fax : 01.58.84.77.50

Coordonnées téléphoniques de l'équipe de direction Hélios :

Jacques MARZIN	Directeur de programme	01.58.84.77.22
Jean Paul BLEHAUT	Directeur Adjoint de programme	01.58.84.77.21
Philippe MAIZY	Directeur Adjoint de programme	01.58.84.75.75
Daniel LEGENDRE	Chef de projet Exploitation	01.30.84.57.51
Jean-Charles MANCEAU	Qualiticien comptable	01.58.84.77.31
Hélène MAOULIDA-MONNIER	Chargée de communication	01.58.84.77.21
Monique GUERINON	Chef de projet Informatique Hélios	01.58.84.77.13
Jean Louis PUELL	Chef de projet Informatique Hélios	01.58.84.77.14

## **12. Annexe A :**

---

H1\_3\_ET\_DOSTEC\_040433\_SystèmeEchangesDonnéesPES\_version1.doc

## **13. Annexe B**

---

Schéma\_PES\_version1.zip

Exemple\_PES\_version1.zip

FIN DU DOCUMENT
-----------------