

VITAM - Mise à jour des unités archivistiques

Version 10.0.0

VITAM

nov. 15, 2026

Table des matières

1 Mise à jour d'un SIP **1**
1.1 Principe de fonctionnement . . . 1
1.1.1 Update . . . 1
1.1.2 Evolutions futures . . . 1
1.2 Mise à jour du schéma XSD SEDA . . . 2
1.2.1 Précision du SystemId . . . 2
1.2.2 Mise à jour de contenu . . . 3
1.2.2.1 Suppression de lien . . . 4
1.2.2.2 Ajout de lien . . . 5
1.3 Exemples de mise à jour . . . 9
1.3.1 Initialisation . . . 9
1.3.2 Mise à jour de contenu simple . . . 9
1.3.3 Ajout d'un lien . . . 9
1.3.4 Suppression d'un lien . . . 10
1.4 Mise à jour via les IHM . . . 10

CHAPITRE 1

Mise à jour d'un SIP

Cette partie décrit la mise à jour d'une Unité Archivistique dans Vitam.

1.1 Principe de fonctionnement

1.1.1 Update

Pour le moment, seul le cas de l'ajout d'une nouvelle Unité Archivistique à une Unité Archivistique existante est en place.

1. Lors de l'extraction SEDA, si une Unité Archivistique déclare un identifiant Vitam dans une Unité Archivistique (*<SystemId>GUID</SystemId>*), alors les données de l'Unité Archivistique existante sont récupérées et un flag *<existing>true</existing>* est ajouté pour indiquer qu'il s'agit d'un mise à jour.
2. Les règles de gestion sont recalculées pour les Unités Archivistiques dont les métadonnées on été modifiées
3. Lors de l'indexation des metadonnées, si l'Unité Archivistique est indiquée comme existante, alors on opère une mise à jour des données et on ajoute l'identifiant de l'opération en cours. Ajourd'hui l'implémentation de l'algorithme prends également en compte la mise à jour des métadonnées : toutes les métadonnées déclarées dans le bordereau s'ajoutent ou remplacent les métadonnées existantes.
4. Alimentation des registres de fond : se comporte comme pour un ajout d'Unité Archivistique pour les existants

1.1.2 Evolutions futures

Toute Unité Archivistique déclarée dans le bordereau sera mise à jour en base car il est obligatoire de déclarer le *<Title>..</Title>* et le *<DescriptionLevel>..</DescriptionLevel>*, il n'est donc pas possible de déclarer que les métadonnées des Unités Archivistiques ne doivent pas être modifiées. Une évolutions du schéma xsd seda sera faite.

De plus, il faut définir une tâche dans le step de vérification du borderau pour apporter une information explicitant quand même la mise à jour. Elle pourrait correspondre techniquement à la vérification de l'existence de l'unité archivistique et pourrait s'intituler « Vérification de l'existence de l'unité archivistique en vue d'une mise à jour ».

1.2 Mise à jour du schéma XSD SEDA

Cette partie a pour but de présenter l'implémentation qui a été mise en place pour enrichir la norme SEDA.

Le but étant de proposer la mise à jour d'UA via le langage XML.

Les différents types de mises à jour :

- Premier type de mise à jour : mise à jour de contenu (der métadonnées ou encore de règles de gestion). Il pourra être question de modifications de contenu (mise à jour / enrichissement de titre, description, ou ajout de nouvelles métadonnées, etc.). On peut déterminer d'ores et déjà que l'on pourra proposer deux modes de mise à jour : un mode PATCH (modification d'un ou plusieurs champs particuliers) et un mode FULL (annule et remplace).
- Deuxième type de mise à jour : la mise à jour de liens. Il s'agit ici de modification (ajout / suppression) de liens entre plusieurs UA. Pour respecter la logique SEDA, les modifications (ajouts et suppressions) de liens se feront toujours du père vers les fils (elles seront déclarées dans l'UA père).

Pour pouvoir décrire ce genre d'opérations via un xml, il convient donc d'étendre le schéma SEDA existant. Pour se faire, il a fallu implémenter un élément abstract : OtherManagementAbstract à l'intérieur de la balise Management pour un unité archivistique donné.

L'implémentation choisie porte le nom « UpdateOperation ». Pour toute opération de mise à jour, il faudra donc préciser une balise « UpdateOperation ».

```
<ArchiveUnit>
    <!-- ADDITIONNAL INFORMATION -->
    <Management>
    <!-- ADDITIONNAL INFORMATION -->
      <UpdateOperation>
         <!-- ADDITIONNAL INFORMATION -->
      </UpdateOperation>
    </Management>
<ArchiveUnit>
```

1.2.1 Précision du SystemId

Pour pouvoir indiquer simplement au moteur VITAM que l'unité archivistique que l'on tente de mettre à jour est déjà existante dans le système, un champ SystemId doit être indiqué. Pour le moment, l'identification d'un archive unit pré-existant dans le système Vitam se fera via son GUID. Plus tard, on étudiera la possibilité de passer par un système de requête pour sélectionner un UA.

```
<ArchiveUnit>
    <!-- ADDITIONNAL INFORMATION -->
    <Management>
    <!-- ADDITIONNAL INFORMATION -->
      <UpdateOperation>
         <SystemId>GUID_MY_ARCHIVE_UNIT</SystemId>
      </UpdateOperation>
    </Management>
<ArchiveUnit>
```

On utilise toujours, au sein du SEDA, les « XML ID » afin de définir des liens entre plusieurs UA. Le système est capable de récupérer les objets concernés grâce à la balise SystemId (id interne à VITAM) s'ils existent déjà dans le SAE. Ainsi, pour créer un lien entre deux UA existants dans le système, il faudra les déclarer tous les deux dans le SEDA via lla balise « XML ID » tout en indiquant le SystemID (GUID).

1.2.2 Mise à jour de contenu

En plus de l'ajout de l'information sur le SystemId pour indiquer au système que l'UA existe déjà, si l'on souhaite effectuer une mise à jour de son contenu, il conviendra, en plus de la précision des données modifiées (dans la partie Content, pas de modification, de ce côté) préciser le type de mise à jour souhaité : totale ou partielle. Une balise FullUpdate permettra de spécifier le type de mise à jour.

Remplacer FullUpdate (false/true) par updateMode (NO_DATA, PATCH, FULL)

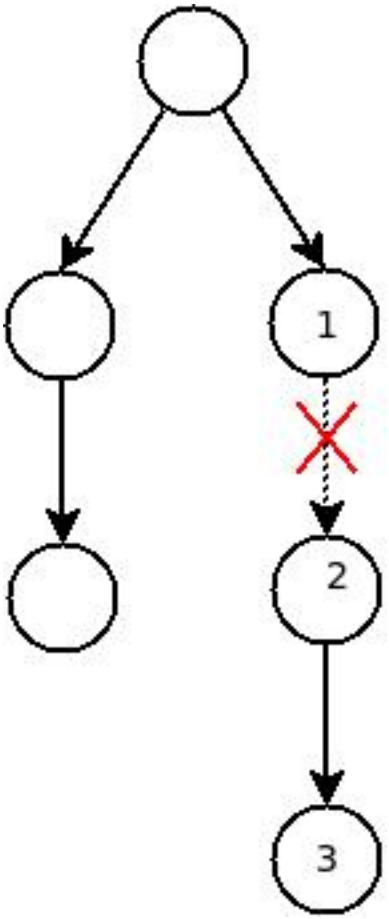
Pour une mise à jour partielle (seuls les champs précisés dans Content seront mis à jour) :

```
<ArchiveUnit>
    <!-- ADDITIONNAL INFORMATION -->
    <Management>
      <!-- Nouvelle regle -->
      <DisseminationRule>
         <Rule>DIS-00001</Rule>
         <StartDate>2008-07-14</StartDate>
      </DisseminationRule>
      <UpdateOperation>
         <SystemId>GUID_MY_ARCHIVE_UNIT</SystemId>
         <FullUpdate>false</FullUpdate>
      </UpdateOperation>
    </Management>
    <Content>
      <!-- Nouvelle titre -->
      <Title>Mon nouveau Titre</Title>
    </Content>
<ArchiveUnit>
```

Pour une mise à jour complète (annule et remplace) :

```
<ArchiveUnit>
    <!-- ADDITIONNAL INFORMATION -->
    <Management>
      <DisseminationRule>
         <Rule>DIS-00001</Rule>
         <StartDate>2008-07-14</StartDate>
      </DisseminationRule>
      <UpdateOperation>
         <SystemId>GUID_MY_ARCHIVE_UNIT</SystemId>
         <FullUpdate>true</FullUpdate>
      </UpdateOperation>
    </Management>
    <Content>
      <DescriptionLevel>Item</DescriptionLevel>
      <Title>Histoire de la station de sa cération à 1946.pdf</Title>
      <TransactedDate>2015-12-04T09:02:25</TransactedDate>
    </Content>
<ArchiveUnit>
```

1.2.2.1 Suppression de lien



La suppression d'un lien entre un UA père et un UA fils sera obligatoirement déclarée sur l'UA père, pour respecter la logique SEDA. Une balise ToDelete permettra de lister les liens entre l'UA père et ses UA fils référencés.

Pour la suppression d'un lien :

```
<ArchiveUnit id="ID_PERE">
    <!-- ADDITIONNAL INFORMATION -->
    <Management>
    <!-- ADDITIONNAL INFORMATION -->
      <UpdateOperation>
         <SystemId>GUID_ARCHIVE_UNIT_PERE</SystemId>
         <ToDelete>
            <ArchiveUnitRefId>XML_ID_FILS_1</ArchiveUnitRefId>
         </ToDelete>
      </UpdateOperation>
    </Management>
<ArchiveUnit>

<ArchiveUnit id="ID_FILS_1">
   <Management>
    <!-- ADDITIONNAL INFORMATION -->
      <UpdateOperation>
         <SystemId>GUID_ARCHIVE_UNIT_FILS_1</SystemId>
      </UpdateOperation>
```

(suite sur la page suivante)

(suite de la page précédente)

```
    </Management>
<ArchiveUnit>
```

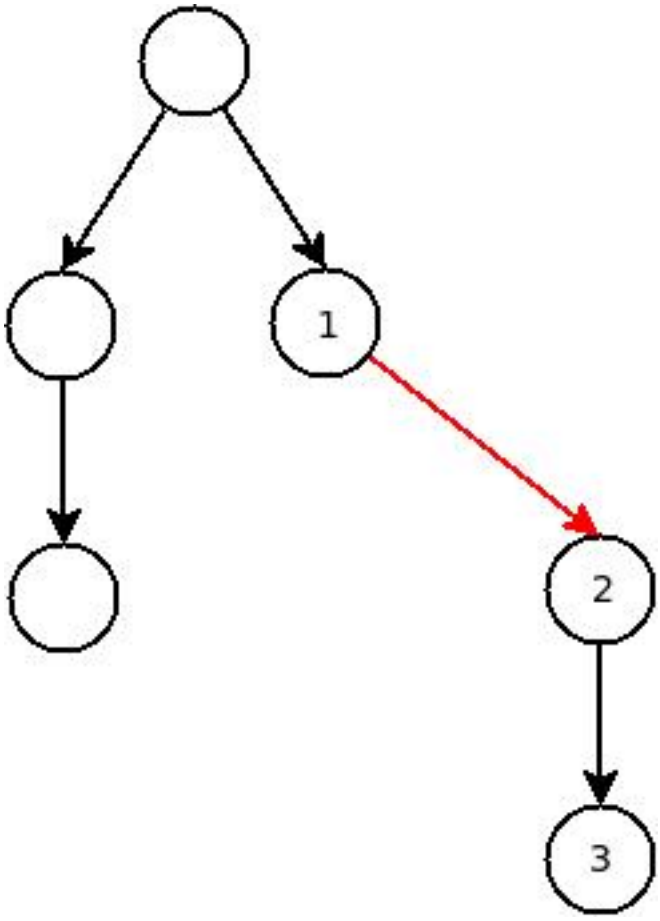
A l'intérieur de la balise ArchiveUnitRefId, on doit référencer un XML ID. C'est à dire un ID interne au xml. Il ne s'agit donc pas ici d'un GUID référencé dans le système Vitam, mais bien une référence à un UA déclaré dans le bordereau. Dans le bordereau doit donc être précisé également le SystemId de l'UA fils référencé, comme indiqué dans l'exemple ci-dessus, sinon le xml ne sera pas valide.

1.2.2.2 Ajout de lien

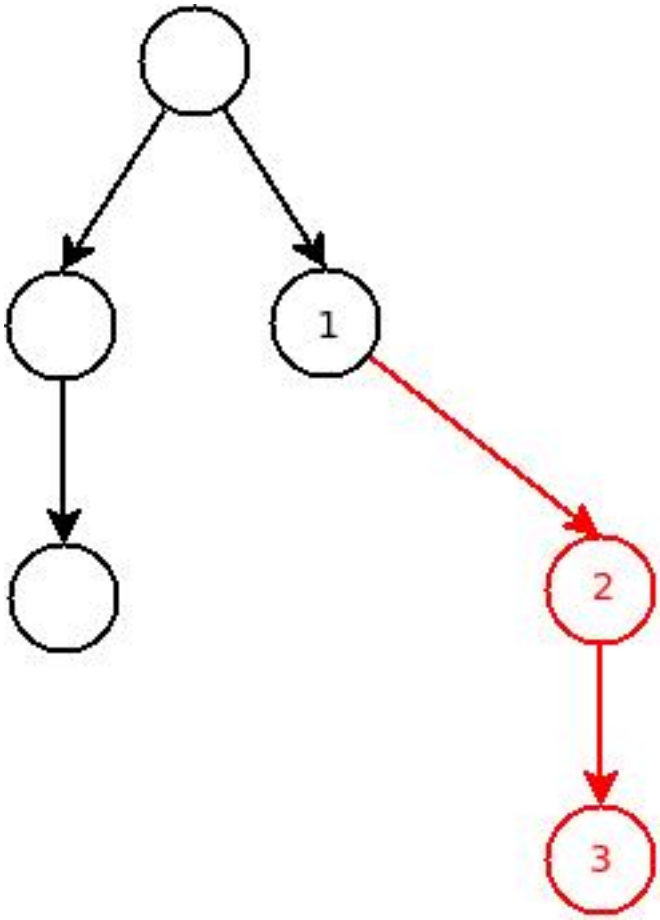
L'ajout de lien entre un UA père et un UA fils sera obligatoirement déclaré sur l'UA père, pour respecter la logique SEDA. En complément de l'utilisation de la nouvelle balise SystemId, il conviendra d'utiliser la balise existante prévue par la norme SEDA : ArchiveUnitRefId.

Quatre cas sont possibles :

- Ajout d'un lien entre deux UA existants déjà dans le système VITAM.



- Ajout d'un nouvel UA fils à un UA père déjà existant dans le système VITAM.



- Ajout d'un UA fils à un nouvel UA père non existant dans le système VITAM.



- Ajout d'un nouvel UA fils à un nouvel UA père.

Pour le cas 1 (Ajout d'un lien entre deux UA existants déjà dans le système VITAM) :

```
<ArchiveUnit id="ID_PERE">
    <!-- ADDITIONNAL INFORMATION -->
    <Management>
    <!-- ADDITIONNAL INFORMATION -->
      <UpdateOperation>
         <SystemId>GUID_ARCHIVE_UNIT_PERE</SystemId>
      </UpdateOperation>
    </Management>
    <ArchiveUnitRefId>ID_FILS_1</ArchiveUnitRefId>
<ArchiveUnit>

<ArchiveUnit id="ID_FILS_1">
   <Management>
    <!-- ADDITIONNAL INFORMATION -->
      <UpdateOperation>
         <SystemId>GUID_ARCHIVE_UNIT_FILS_1</SystemId>
      </UpdateOperation>
    </Management>
<ArchiveUnit>
```

Pour le cas 2 (Ajout d'un nouvel UA fils à un UA père déjà existant dans le système VITAM) :

```
<ArchiveUnit id="ID_PERE">
    <!-- ADDITIONNAL INFORMATION -->
    <Management>
```

(suite sur la page suivante)

(suite de la page précédente)

```
    <!-- ADDITIONNAL INFORMATION -->
      <UpdateOperation>
         <SystemId>GUID_ARCHIVE_UNIT_PERE</SystemId>
      </UpdateOperation>
    </Management>
    <ArchiveUnitRefId>ID_FILS_1_NOUVEAU</ArchiveUnitRefId>
<ArchiveUnit>

<ArchiveUnit id="ID_FILS_1_NOUVEAU">
   <Management>
      <!-- Information sur le management -->
   </Management>
   <Content>
      <!-- Information sur le content -->
   </Content>
<ArchiveUnit>
```

Pour le cas 3 (Ajout d'un UA fils à un nouvel UA père non existant dans le système VITAM) :

```
<ArchiveUnit id="ID_PERE">
   <Management>
      <!-- Information sur le management -->
   </Management>
   <Content>
      <!-- Information sur le management -->
   </Content>
   <ArchiveUnitRefId>ID_FILS_1_EXISTANT</ArchiveUnitRefId>
<ArchiveUnit>
<ArchiveUnit id="ID_FILS_1_EXISTANT">
    <!-- ADDITIONNAL INFORMATION -->
    <Management>
    <!-- ADDITIONNAL INFORMATION -->
      <UpdateOperation>
         <SystemId>GUID_ARCHIVE_UNIT_FILS_1_EXISTANT</SystemId>
      </UpdateOperation>
    </Management>
<ArchiveUnit>
```

Pour le cas 4 (Ajout d'un nouvel UA fils à un nouvel UA père - cas nominal) :

```
<ArchiveUnit id="ID_PERE">
   <Management>
      <!-- Information sur le management -->
   </Management>
   <Content>
      <!-- Information sur le management -->
   </Content>
   <ArchiveUnitRefId>ID_FILS_1_NOUVEAU</ArchiveUnitRefId>
<ArchiveUnit>
<ArchiveUnit id="ID_FILS_1_NOUVEAU">
   <Management>
      <!-- Information sur le management -->
   </Management>
   <Content>
      <!-- Information sur le management -->
   </Content>
<ArchiveUnit>
```

1.3 Exemples de mise à jour

Vous trouverez ci-dessous des exemples d'utilisation (à adapter, bien évidemment) d'utilisation des différentes opérations de mise à jour.

1.3.1 Initialisation

Pour pouvoir effectuer des opérations de mise à jour, il convient d'effectuer une Entrée. Pour notre exemple, nous allons réaliser l'import d'un SIP renseignant :

- 1 ArchiveUnit XML_ID1.
- 1 ArchiveUnit XML_ID2 seul.
- 1 ArchiveUnit XML_ID_FILS1 rattaché à l'ArchiveUnit XML_ID1. Cet ArchiveUnit référençant un DataObject, déclaré dans le bordereau.
- 1 ArchiveUnit XML_ID_FILS2 rattaché à l'ArchiveUnit XML_ID1.

Le SIP d'initialisation se trouvant ici : :download :"<files/SIP_INIT.zip>" Le bordereau : :download :"<files/manifest_INIT.xml>"

1.3.2 Mise à jour de contenu simple

Le but ici est de mettre à jour les métadonnées de l'ArchiveUnit XML_ID1 d'origine. Le bordereau permettant de faire cette mise à jour : :download :"<files/manifest_UPDATE_CONTENT.xml>"

Ci-dessous un extrait de la syntaxe :

```
<UpdateOperation>
      <SystemId>aeaaaaaaaaaam7mxab2kkakzn5mib7aaaaaq</SystemId>
      <FullUpdate>false</FullUpdate>
</UpdateOperation>
```

Note : il conviendra de remplacer le contenu de la balise <SystemId> par l'identifiant interne généré lors de l'entrée original.

Après l'import du SIP contenant ce bordereau (pas d'objet nécéssaire dans le SIP), les informations de l'ArchiveUnit XML_ID1 seront mises à jour.

1.3.3 Ajout d'un lien

Le but ici est l'ajout d'un lien entre l'ArchiveUnit XML_ID2 et l'ArchiveUnit XML_ID_FILS2. Le bordereau permettant de faire cette mise à jour : :download :"<files/manifest ADD_LINK.xml>"

Ci-dessous un extrait de la syntaxe :

```
<!-- Dans la balise Management de l'ArchiveUnit XML_ID2 -->
<UpdateOperation>
   <SystemId>aeaaaaaaaaaam7mxab2kkakzn5micdiaaaaq</SystemId>
</UpdateOperation>
<!-- Puis plus loin toujours dans la balise ArchiveUnit de XML_ID2 -->
<ArchiveUnit id="XML_ID21">
     <ArchiveUnitRefId>XML_ID_FILS2</ArchiveUnitRefId>
 </ArchiveUnit>
```

Note : il conviendra de remplacer les contenus des balises <SystemId> par les identifiants internes générés lors de l'entrée original.

Après l'import du SIP contenant ce bordereau (pas d'objet nécéssaire dans le SIP), l'ArchiveUnit XML_ID2 sera un père de l'ArchiveUnit XML_ID_FILS2.

1.3.4 Suppression d'un lien

Le but ici est la suppression d'un lien existant entre l'ArchiveUnit XML_ID1 et l'ArchiveUnit XML_ID_FILS2. Le bordereau permettant de faire cette mise à jour : :download :"<files/manifest DELETE_LINK.xml>"

Ci-dessous un extrait de la syntaxe :

```
<UpdateOperation>
   <SystemId>aeaaaaaaaaaam7mxab2kkakzn5mib7aaaaaq</SystemId>
   <ToDelete>
      <ArchiveUnitRefId>XML_ID_FILS2</ArchiveUnitRefId>
   </ToDelete>
</UpdateOperation>
```

Note : il conviendra de remplacer les contenus des balises <SystemId> par les identifiants internes générés lors de l'entrée original.

Après l'import du SIP contenant ce bordereau (pas d'objet nécéssaire dans le SIP), l'ArchiveUnit XML_ID1 ne sera plus un père de l'ArchiveUnit XML_ID_FILS2.

1.4 Mise à jour via les IHM

En ce qui concerne l'utilisation via les IHM minimales, ce n'est pas encore possible. Cette évolution sera dévelopée lors d'une unser story associée.