

ANNEXE n°4

Présentation des principaux workflows Vantage et de leurs fonctions.

Compilation

Le workflow compilation est traité par 4 jobs qui utilisent les briques suivantes : Watch, Workorder, Analyse, Construct, Compute, Decide , Notify, Populate, Associate, Flip64, Move , Delete, Message.

Il s'appuie sur 10 API, déclenchées par Notifiy via Invoke.

Le dépôt d'un fichier EDL accompagné de ses médias associés déclenche le processus.
Le watch surveille plusieurs répertoires, chacun déclenchant un traitement différent.
Ces traitements sont différenciés via un module de décision qui s'appuie sur le nom du dossier de dépôt pour déterminer la nature du traitement à appliquer.

Un premier processus analyse les médias et, si nécessaire, les transcode au bon format (par exemple, en 16/9).

En parallèle, l'EDL déclenche un second processus qui attend la fin du traitement des médias.

Selon la provenance du dépôt, ce processus peut :

- Concaténer les extraits souhaités
- Extraire les extraits individuellement (mode "extrait")

Le mode extrait s'appuie sur le fichier concaténé pour conserver une continuité des timecodes, ce qui assure une traçabilité.

Enfin :

- Le dépôt est purgé en différé
- Les éléments nécessaires (EDL, etc.) sont archivés

Conformation Directe

Le workflow conformation directe est traité par 4 jobs qui utilisent les briques suivantes : Watch, Identify, Workorder, Construct, Compute, Decide , Notify, Populate, Flip64, Move , Delete, Message.

Il s'appuie sur 5 API, déclenché par Notifiy via Invoke.

Le dépôt d'une EDL déclenche le processus.

Les étapes sont les suivantes :

1. Les informations de l'EDL sont traitées via les modules Identify, Compute et Construct.
2. Une API externe est appelée (via un module Invoke dans Notify) pour vérifier les données.
3. Après extraction des noms de fichiers depuis l'EDL :
 - On vérifie leur présence dans l'emplacement cible

- On s'assure que les fichiers ne sont plus en cours de copie

Le résultat de l'API est stocké dans une variable résultat.

Un module « décide » analyse ce résultat :

- En cas d'erreur : le processus est mis en erreur avec envoi d'une notification explicative
- Si tout est correct : le processus continue

Une seconde API vérifie la cohérence des timecodes (TC) entre l'EDL et les médias.

Exemple : si le TC OUT dans l'EDL est 01:00:00:00 mais le média à 00:30:00:00, une erreur est levée.

Si la vérification est concluante :

- Les médias souhaités sont extraits
- Puis déposés dans un dossier de livraison

Enfin :

- Le dépôt est purgé en différé
- Les éléments nécessaires (EDL, etc.) sont archivés

Conformation

Le workflow conformation est traité par 2 jobs qui utilisent les briques suivantes : Watch, Identify, Workorder, Construct, Compute, Decide , Notify, Analyze, Flip64, Move , Delete, Message.

Il s'appuie sur 3 API impulsé par Notifiy via Invoke.

Une EDL déclenche ce processus. Un watch surveille plusieurs dossiers.

Une arborescence de décisions permet de :

- Récupérer le chemin du dépôt
- Déterminer le traitement à appliquer

Une API vérifie la présence et l'état des médias listés.

Selon les résultats, un système de « décide » :

- Oriente vers une mise en erreur avec notification, ou
- Permet de poursuivre le processus

Selon le chemin de dépôt, plusieurs types de transcodage peuvent être attendus :

- Logo couleur
- Logo noir et blanc
- Timecode visible, etc.

Un fichier CSV est généré et déclenche un processus parallèle qui :

- Transcode les médias selon les paramètres attendus

Transcodage par nommage de dossier

1. Transcodage par codec avec options (ex. H264)

Ce workflow est traité par 5 jobs qui utilisent les briques suivantes : Watch, Identify, Forward, Receive, Construct, Compute, Decide, Flip64, Move , Delete.

Le nom du dossier de dépôt détermine les traitements à effectuer, selon des mots-clés prédéfinis.

Exemple :

Dossier EN_H264_logo_nb_tc_XDCAM →

Le fichier déposé sera transcodé en :

- H264
- Avec logo noir et blanc
- Avec timecode visible
- En XDCAM

2. Transcodage H264 avec paramètres personnalisés

Ce workflow est traité par 1 job qui utilise les briques suivantes : Watch, Identify, Construct, Compute, Decide , Analyze, Flip64, Move , Delete.

Une syntaxe normalisée dans le nom du dossier encode les paramètres de sortie.

Exemple :

EN_H264_1920x1080_4Mb_25p_1P_TC_logo

→ Paramètres extraits :

- Codec : H264
- Résolution : 1920x1080
- Débit : 4 Mb/s
- Fréquence : 25p
- GOP : 1P
- Affichage du TC : Oui
- Logo : Oui

Import dans Interplay

Ce workflow est traité par 9 jobs qui utilisent les briques suivantes : Watch, Identify, Analyze, Forward, Receive, Media Creation, Construct, Compute, Decide, Flip64, Move , Delete.

Il s'appuie sur 1 API impulsé par Notifiy via Invoke.

Préanalyse

Avant l'import, les fichiers sont analysés pour exclure ceux de moins de 3 secondes.

Dépôt

- **Dépôt simple :**
Un seul fichier → livré dans un dossier structuré année/date_du_jour
- **Dépôt multiple :**
Plusieurs fichiers → livrés dans année/date_du_jour/nom_du_dossier_de_dépôt, un processus vient compléter la purge par la suppression des dossiers de dépôt vide.

Import Vidéos

Ce workflow est géré par 8 jobs utilisant les briques suivantes :

Watch, Analyse, Associate, Populate, Forward, Receive, Identify, Extract, Construct, Compute, Notify, Copy, Decide, Flip64, Move, Delete, Message.

Il s'appuie sur 9 appels API déclenchés par Notify via le module Invoke.

Déroulé du processus

1. **Déclenchement**
 - Le dépôt d'un fichier vidéo déclenche le processus.
 - Le job de départ s'appuie sur trois "watch" différents, qui modifient le traitement en fonction de la provenance du fichier.
2. **Analyse et vérification**
 - Un checksum est effectué sur les fichiers MXF déjà au bon format.
 - Si le fichier est en AVI, un transcodage est lancé vers le format attendu, suivi d'un checksum du fichier résultant.
3. **Gestion des proxys**
 - Les fichiers déposés peuvent être accompagnés d'un proxy, avec plusieurs conventions de nommage possibles.
 - Le processus :
 - Vérifie la présence des proxys
 - Les renomme correctement si nécessaire
 - Si un proxy est manquant ou que sa durée ne correspond pas au fichier noble, il est recréé
 - Un proxy "web" est également généré.
4. **Archivage et contrôle d'intégrité**
 - Une fois le chemin de dépôt reconstitué, les médias sont archivés.
 - Un nouveau checksum est calculé et comparé au précédent pour vérifier l'intégrité du fichier maître.
5. **Journalisation**
 - Les étapes clés du processus sont enregistrées dans un fichier JSON, qui regroupe l'ensemble des traitements de la journée.
 - Ce fichier JSON est archivé chaque soir.
6. **Suppressions**
 - Le dépôt est purgé.

Notifications

- Des notifications automatiques sont envoyées :
 - Au dépositaire
 - À une boîte mail générique

Ces messages informent des succès ou des échecs du processus

Import Photos

Ce workflow est géré par deux jobs, qui s'appuient sur les briques suivantes : Watch, Identify, Extract, Construct, Compute, Notify, Decide, Flip64, Move, Delete, Message.

Il repose sur 7 appels API déclenchés via la brique Notify utilisant le module Invoke.

Déroulé du processus

1. **Déclenchement**
 - Le dépôt d'un fichier XML déclenche le processus.
 - Un scan du dossier de dépôt est effectué pour détecter la présence des médias associés.
2. **Vérification documentaire**
 - Une requête est envoyée à un serveur distant pour vérifier l'existence d'une fiche documentaire correspondant au contenu déposé.
 - En retour, le serveur fournit le chemin de livraison attendu.
3. **Validation des médias**
 - Une vérification est faite sur le serveur de stockage pour s'assurer que les médias ne sont pas déjà présents.
4. **Dépôt**
 - Si tout est conforme, les médias sont déposés à l'emplacement indiqué.
 - Un fichier récapitulatif est généré, contenant un numéro de compteur unique, puis déposé dans une GED "import".
5. **Journalisation**
 - Les étapes clés du traitement sont enregistrées dans un fichier JSON regroupant les processus de la journée.
 - Ce fichier JSON est archivé chaque soir.
6. **Suppressions**

Le dépôt est purgé.

Notifications

- Des notifications automatiques sont envoyées :
 - Au dépositaire
 - À une boîte mail générique
- Ces notifications informent des succès ou des erreurs rencontrés durant le processus.

Versement Photo

Ce workflow est géré par 2 jobs, qui utilisent les briques suivantes :
Watch, Identify, Extract, Construct, Compute, Notify, Copy, Decide, Flip64, Move, Delete, Message.

Il repose sur 9 appels API, déclenchés par Notify via le module Invoke.

Déroulé du processus

1. Déclenchement

- Le dépôt d'un fichier XML déclenche automatiquement le processus.

2. Vérification des médias

- Une vérification est effectuée sur le serveur de stockage pour s'assurer que les médias listés dans le XML ne sont pas déjà présents.
- Les médias déposés sont ensuite contrôlés :
 - Présence effective des fichiers
 - Comparaison du poids avec les données fournies dans le XML

3. Archivage

- Une fois validés, les fichiers sont déposés sur le serveur d'archivage.
- Un fichier récapitulatif est généré avec un numéro de compteur unique. Il est déposé dans une GED "versement".

4. Journalisation

- Les étapes clés du traitement sont enregistrées dans un fichier JSON retraçant l'ensemble des processus de la journée.
- Ce fichier JSON est archivé chaque soir.

5. Suppressions

- Le dépôt est purgé.

Notifications

- Des notifications automatiques sont envoyées :
 - Au dépositaire
 - À une boîte mail générique
- Elles informent du succès ou de l'échec du processus.

Versement Vidéos

Ce workflow est géré par 4 jobs, qui utilisent les briques suivantes :
Watch, Identify, Extract, Analyse, Populate, Associate, Forward, Receive, Construct, Compute, Notify, Copy, Decide, Flip64, Move, Delete, Message.

Il repose sur 10 appels API, déclenchés par Notify via le module Invoke.

Déroulé du processus

1. Déclenchement

- Le dépôt d'un fichier XML déclenche automatiquement le processus.

2. Identification du type de versement

- Une identification du type de versement est réalisée à partir du XML :
 - **PM** (Production Monté)
 - **TO** (Rushes)

3. Vérification des médias

- À partir du contenu du XML :
 - On vérifie que les médias listés ne sont pas déjà présents sur l'espace d'archivage.
 - On contrôle la présence effective des fichiers et la conformité de leurs caractéristiques (poids, nommage, etc.) avec les informations du XML.

4. Concaténation et création de proxys

- Les médias listés sont concaténés en un seul fichier.
- Deux types de proxys sont générés :
 - Un proxy standard
 - Un proxy web

5. Archivage

- Le chemin d'archivage et le nommage du fichier est construit dynamiquement selon plusieurs critères :
 - Année de création
 - Date de dépôt
 - Nom du fichier
 - Nom de l'organisme versant, etc.
- Ces fichiers sont ensuite livrés dans le stockage d'archivage.
- Un fichier récapitulatif est généré avec un numéro de compteur unique, puis déposé dans une GED "versement".

6. Journalisation

- Les étapes clés du traitement sont enregistrées dans un fichier JSON récapitulant les processus de la journée.
- Ce fichier JSON est archivé chaque soir.

7. Suppressions

Le dépôt est purgé.

Notifications

- Des notifications automatiques sont envoyées :
 - Au dépositaire
 - À une boîte mail générique

Ces notifications informent des succès ou des échecs du processus.