

AUTOMATOR TRAVAILLE POUR VOUS

Automatisez sous Mac



SYLVAIN GAMEL

Préface



Mentions légales

Conception de la couverture : Sylvain GAMEL

Illustrations photographiques : Sylvain GAMEL

ISBN (iBook Store) : 978-2-9548792-0-8

Prix (iBook Store) : 12,99 euros

© Sylvain GAMEL, 2014

Les programmes figurant dans ce livre ont pour but d'illustrer les sujets traités. Il n'est donné aucune garantie quant à leur fonctionnement une fois compilés, assemblés ou interprétés dans le cadre d'une utilisation professionnelle ou commerciale.

Toute représentation ou reproduction, intégrale ou partielle, faite sans le consentement de l'auteur, de ses ayants droit, ou ayants cause, est illicite (loi du 11 mars 1957, alinéa 1er de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal. La loi du 11 mars 1957 autorise uniquement, aux termes des alinéas 2 et 3 de l'article 41, les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective d'une part et, d'autre part, les analyses et les courtes citations dans un but d'exemple et d'illustration.

Les informations diffusées n'engagent pas la responsabilité de l'éditeur ou de l'auteur.

Les marques citées appartiennent à leurs propriétaires respectifs.

Préface

En 1998, lors d'une de ses fameuses keynotes, Steve JOBS fait monter sur scène Sal SOGOHIAN, chef produit AppleScript, pour présenter les capacités de communication inter-applications et d'automatisation de Mac OS 8.5 à l'aide du langage de scripting AppleScript intégré au système depuis quelques années. Avec quelques lignes d'un code proche de l'anglais, une application transfère automatiquement des fiches d'une base de données de petites annonces pour générer à la volée un catalogue dans le logiciel Quark XPress. Alors qu'un autre script colorie les pays d'une carte des États-Unis d'Amérique en adaptant la couleur en fonction de la température... récupérée en temps réel sur Internet ! Cette démonstration époustouflante avait fait forte impression, et m'a longtemps marquée par son efficacité et sa simplicité... apparente. Car si le langage Applescript permet d'automatiser des tâches, son approche n'est pas forcément accessible à tout un chacun.

Mais Apple est parfois une société têtue, et rendre des technologies complexes utilisables par tous reste un de ses chevaux de bataille. Et en juin 2004, lors de la première démonstration de Mac OS X 10.4, Sal SOGOHIAN est de retour sur scène pour présenter une nouvelle solution d'automatisation. Et celle-ci est vraiment conçue pour tout le monde (« *the rest of us* »), comme aime le dire malicieusement Apple. Qui n'a jamais senti sa motivation fléchir au moment de

renommer une centaine de fichiers avec la date du jour ? Ou de devoir convertir des fichiers d'un format JPEG vers TIFF, ou de les recadrer à une certaine taille ? Ou encore de devoir télécharger tous les fichiers intégrés dans une page web ? J'ai par exemple eu une fois une requête particulière d'un de mes clients : comment convertir chaque page d'un document PDF de deux-cent pages en fichiers PNG indépendants ? Vous conviendrez que faire une copie d'écran par page peut être légèrement frustrant...

[Automator](#) est la réponse parfaite qu'Apple a composée pour rendre l'automatisation accessible à tous. Cette solution est conviviale, simple à mettre en œuvre, mais regorge de petites subtilités. Mieux encore : [Automator](#) est puissant mais reste un outil fun. Un pur produit Apple, en quelque sorte ! Peut-être serez-vous déçu de découvrir qu'[Automator](#) ne peut pas faire en standard exactement tout ce que vous souhaitez... mais vous verrez cependant grâce à ce livre qu'il peut se plier à bien des besoins.

Finalement, le plus gros manque d'[Automator](#) est sûrement de ne pas indiquer le temps qu'il nous fait gagner chaque année. Cela nous inciterait sûrement à l'utiliser encore beaucoup plus ! Les recettes proposées par Sylvain sont d'ors et déjà une excellente source d'inspiration, mais n'hésitez pas à expérimenter. Automator est un outil rare, qui se découvre au jour le jour... et dont la richesse n'a pas fini de vous épater.

Guillaume GÈTE
blog.gete.net

L'équipe

L'auteur

Développeur depuis 1997, *Sylvain GAMEL* connaît principalement les environnements Unix, Java et Mac OS X. Il pratique notamment les langages C, Objective C, Java et JavaScript.

Depuis 2010, il est développeur indépendant. En plus, de son activité de consultant, il édite également des applications pour iOS (**Calcullette TVA**, Cartes SCRUM).

Il anime le site **Automatisez.net** et tient également un blog où il partage astuces et connaissances autour du Mac, de l'iPhone et l'informatique en général. Un peu hyperactif des réseaux sociaux vous pouvez le trouver sur **Tumblr** ou Twitter (**@sgamel**), mais aussi sur **LinkedIn**, **Flipboard** et **Facebook**.

Il a également écrit la mise à jour du livre Objective-C Précis et concis pour la version 2 du langage (éditions Digit Books).

Le préfacier

Guillaume GÈTE est consultant Apple indépendant et formateur, spécialiste de l'intégration Mac OS X et iOS X en entreprise. Auteur de nombreux livres références sur OS X et iOS, il diffuse par ailleurs sur son blog (**blog.gete.net**) astuces et points de vue sur la galaxie Apple (et parfois des actions Automator et Applescript). Actif sur Twitter (**@ggete**), il coanime également depuis 2013 « **3 hommes et un podcast** », un podcast orienté Apple, cinéma et séries télé. Il est le développeur de **Lion Diskmaker**, outil développé en Applescript et maintes fois primé. À ses heures perdues, il aime les jeux vidéo, les Transformers, Doctor Who et jouer à Des chiffres et des lettres.

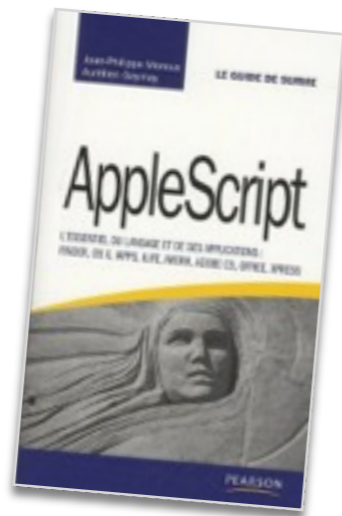


Les relecteurs

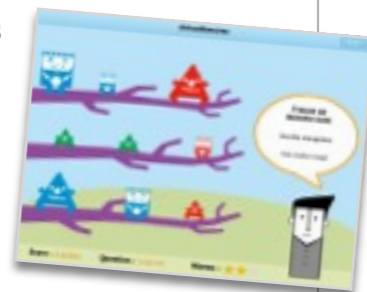
Aurélien GAYMAY est passionné d'informatique, de High-Tech, d'écologie et surtout d'Apple. Il découvre le monde Mac depuis son arrivée dans une « entreprise » dédiée à la messagerie médicale sécurisée donc il est technicien informatique.

C'est aussi grâce à son travail qu'il a découvert les joies de l'AppleScript qui lui permet de simplifier ses journées de travail en automatisant des opérations répétitives et évitant ainsi une perte de temps. L'AppleScript l'a sauvé et il a coécrit l'ouvrage **AppleScript : Guide de survie aux éditions Pearson**.

Il est également responsable de la rubrique Mac et iOS du site **developpez.com** et, à ses heures perdues, pratique un peu de développement web et iOS. Aurélien est également très actif sur Twitter (**@mackortex**).



Jean-Marie LAFON est un membre actif de l'association **EducOO.org**, qui est à l'origine des projets **OOo4Kids** et **OOoLight**, tous les deux dérivés de la célèbre suite bureautique libre **OpenOffice.org**. Il est également l'auteur de la version Mac OS X de l'application **Bréviaire d'Orthographe Française** (**paisible.fr**) ainsi que plusieurs applications pour iOS comme les **Jeux de TonTon**.



Préambule

Ce livre reprend le contenu des deux ouvrages précédemment édités par DigitBooks. Je m'efforce également d'y ajouter quelques éléments spécifiques à la dernière version d'OS X 10.9 Mavericks. Avec ce nouvel ouvrage, mon objectif est avant tout de regrouper le contenu des deux précédents volumes pour en faire un matériel plus cohérent.

Ce livre utilise bien entendu le support écrit, mais une large partie du texte vient en support des multiples vidéos qui détaillent de façon la plus visuelle possible les exemples présentés.

J'espère que vous trouverez ici le maximum d'informations quelque soit votre niveau en informatique.

Pour commencer

« A begining is the time for taking the most delicate care that the balances are correct. »^{}*

Dune. Franck H. HERBERT

** Un commencement est le moment qui convient pour prendre les précautions les plus délicates afin que les équilibres soient respectés.*



Introduction

L'ordinateur est un outil extraordinaire qui a beaucoup changé notre quotidien. Fini les montages de films à coup de ciseaux et de ruban adhésif. Fini l'attente du développement des photos... et découvrir que la moitié est bonne à jeter. Plus de pile de disques à la maison, ni même de lourde machine à écrire.

Cette « *bicyclette pour l'esprit* » ⁽¹⁾ remplace nombre de gadgets et remplit bien plus de rôles qu'on ne l'imaginait au début de l'informatique personnelle.

Si son fonctionnement est bien plus souple que celui d'une tablette, comme l'iPad, il est aussi bien plus complexe et intimidant. Et au fil du temps, vous pourrez avoir l'impression que votre ordinateur vous oblige à réaliser mécaniquement nombre d'actions répétitives.

Si vous avez déjà eu cette impression, alors ce livre est fait pour vous !

Apprenez à laisser votre ordinateur prendre en charge les tâches les plus répétitives pour vous concentrer sur l'essentiel.

Il est temps pour vous de prendre en main votre bicyclette de l'intellect.

Prérequis

La première édition de ce livre a été écrite après la sortie de Mac OS X 10.7 Lion, et avant la sortie d'OS X 10.9 Mavericks. Les copies d'écran et les comportements décrits utilisent donc ces différentes versions du système. Si vous êtes un nouvel utilisateur Mac et que vous n'avez jamais utilisé les versions précédentes du système, cela ne devrait pas vous poser de problèmes. Si vous avez déjà utilisé Mac OS X, vous devriez déjà connaître les petites différences entre Lion et ses prédécesseurs :

- sauvegarde automatique ;
- remplacement d'**Enregistrer sous...** par **Dupliquer** ;
- etc.

Mavericks n'apportera pas beaucoup de changements dans Automator, mais le style visuel de certaines applications est maintenant plus épuré.

En règle générale, le contenu reste applicable à l'identique aux systèmes antérieurs à Lion. Si ce n'est pas le cas, je m'efforce de vous donner une solution alternative.

De la même façon, si certaines fonctionnalités ne sont disponibles qu'à partir d'une certaine version du système je m'efforce de le préciser.

Il n'est pas nécessaire d'avoir des notions de programmation. Nous emploierons des langages de scripts comme AppleScript, Perl ou le Shell, mais ces utilisations restent ponctuelles et sont plus des recettes utilisables telles que. J'espère que vous aurez tout de même envie d'approfondir un peu ces sujets, car connaître d'autres langages vous permettra d'enrichir un peu plus Automator et de repousser ses limites.

Pour la dernière partie de l'ouvrage, il vous faudra pouvoir installer l'environnement de développement du Mac à partir du Mac App Store. Pour cela, vous devez avoir un compte Apple. Les outils étant gratuits, il n'est pas nécessaire de lier un moyen de paiement à ce compte. Pour rappel, un compte iTunes est aussi un compte Apple, vous pouvez donc l'utiliser pour vos téléchargements sur le Mac App Store. Et comme il s'agit de programmation, quelques notions de base sont fortement recommandées.

Conventions typographiques

Dans le texte

Voici les différentes conventions que nous suivons :

- **Menus**, commande, action, élément d'interface et raccourci-clavier sont ainsi signalés.
- Le symbole ⌘ représente la touche commande (« **cmd** »). Votre clavier possède deux de ces touches, une de part et d'autre de la barre d'espace.
- Le symbole ⇧ représente la touche majuscule. Là encore, le clavier contient deux touches majuscules, une de chaque côté du clavier.
- **Application**. Les noms d'application comme le **Finder** ou **Automator**.
- On désigne un nom de fichier de cette façon : **fichier** ou **dossier**.
- *Italique*. Mets en exergue les termes nouveaux ou la signification des acronymes.
- Police à chasse fixe. Mets en valeur les éléments de code dans le texte.

Extraits de code

Les exemples ou extraits de code sont présentés comme suit :

```
-- Commentaire
tell application "Finder"
    set variable to "un texte"
    activate
end tell
```

Les fragments de code contiennent ainsi des **mots clés** du langage ou des *variables*.

Une mise en forme spécifique met en avant les **commentaires** ainsi que les littéraux de *textes*.

Dans un code source, on mettra également en avant les **commandes**, **messages** ou noms de **méthodes**.

Remarques

! Des notes et remarques éveillent votre attention sur des points précis.

... Les points importants sont à lire absolument.

🎯 Le texte ainsi signalé vous met en garde.

Les exemples de cet ouvrage sont disponibles sur Automatisez.net, le site du livre. Ce site sera aussi pour nous, je l'espère, le moyen de garder le contact.

Les exemples fournis au téléchargement sont indiqué par un célèbre robot jaune, comme ici.



Qu'est-ce que l'automatisation ?

Définition : L'automatisation est la mise en œuvre d'un dispositif se comportant de façon répétitive et sans intervention humaine.

Cette définition peut vous sembler un peu vaste, mais elle intègre deux critères importants :

- la répétition ;
- l'absence d'intervention humaine.

Toute action répétitive peut être mise en œuvre dans un automate.

Comme pour une pièce de théâtre, on peut être certain que votre processus automatique déroulera votre script comme autant de représentations. Que vous changiez de salle, d'acteurs ou même de décor, votre pièce de théâtre restera la même et suivra toujours la même trame.

L'absence d'intervention humaine n'est, elle, pas systématique. Autant, vous pourrez construire un processus totalement automatique, autant vous pourrez le rendre hautement interactif. Ce critère-là est à votre discrétion. En automatisant des séquences d'actions, vous allez donc devenir l'auteur de petites pièces de théâtre que votre Mac rejouera pour vous à la demande.

Vos applications et le système de votre Mac seront les personnages de vos pièces.

Dans le cadre de ce livre, lorsque nous parlerons d'automatisation, il s'agira donc de piloter, avec une intervention plus ou moins importante de votre part, des applications et le système OS X.

Et l'outil que vous allez apprendre à utiliser est déjà dans votre Mac. Ce logiciel porte bien son nom : [Automator](#).



IMAGE 1.1 L'icône de l'application Automator

Pourquoi automatiser ?

La première réponse qui vous vient probablement à l'esprit est probablement d'éliminer les tâches manuelles répétitives. C'est une réponse tout à fait valable, mais ce n'est pas la seule.

On peut vouloir créer des procédures automatiques pour la même raison que l'on utilise des logiciels :

- simplifier un peu plus l'utilisation de l'ordinateur ;
- standardiser les procédures d'une entreprise ;
- fournir des outils homogènes aux utilisateurs.

Par exemple, un utilisateur novice ne sait pas forcément comment envoyer par courriel une image trouvée sur Internet. Il saura peut-être sauvegarder cette image sur le bureau par un **glisser-déposer**. Vous pouvez ainsi fournir à cet utilisateur un outil capable de voir les nouveaux fichiers du bureau pour proposer spontanément à l'utilisateur de l'envoyer dans un message. [Automator](#) vous donne les outils pour tout mettre en place. Si vous êtes un minimum versé dans l'informatique vous êtes probablement aussi le support technique de votre cercle familial. [Automator](#) pourra vous aider à simplifier la vie de votre famille.

Gestion de documents ?

Une société souhaite classer et archiver ses documents de façon homogène sans pour autant recourir à une application de gestion documentaire. [Automator](#) peut fournir les outils qui garantissent un traitement cohérent des fichiers.

Photo ?

Vous faites de la photo numérique et vous aimez gérer vos fichiers image avec rigueur : nom utilisant la date, rangement dans des dossiers par date, par pellicule, etc. Renommer les fichiers et les ranger vous prend un temps fou ? [Automator](#) peut vous aider et faire cela à votre place.

Vous pensiez qu'un logiciel complexe est nécessaire pour tout cela ? Vous vous trompez. Mac OS X vous donne une palette complète d'outils pour mettre en œuvre ce genre de solution. Automator n'est que le plus simple et abordable d'entre eux, autant en profiter et apprendre à l'utiliser.

Les domaines d'application d'un processus automatique sont aussi vastes que votre imagination.

Comment automatiser ?

L'idée d'automate

Les automates que vous allez réaliser sont de véritables petits logiciels. Sur le fond, cela relève de la programmation informatique. Mais ne vous effrayez pas, car, en pratique, ces « logiciels » resteront très simples, tout aussi simples et abordables que les outils que vous allez utiliser.

Un processus automatique est construit comme le dialogue d'une pièce de théâtre.

On identifie le personnage qui doit parler. On lui donne son texte. On passe à la réplique suivante qui identifie le personnage, donne son texte. Et ainsi de suite jusqu'au mot « fin ».

Toute cette phase de rédaction du scénario se déroule dans un outil particulier, entièrement spécialisé pour cette tâche : *l'éditeur Automator*.

Vos personnages sont des logiciels : applications et briques du système d'exploitation OS X. Les dialogues sont des actions comprises par ces logiciels. Vous identifiez les acteurs de la pièce, indiquez quelles répliques chacun doit prononcer et dans quel ordre.

Comme je l'ai déjà dit, [Automator](#) n'est pas le seul outil à votre disposition. Vous pourrez en utiliser d'autres comme AppleScript, les scripts Shell Unix et même le langage de programmation Objective-C. Le choix des armes sera le vôtre et dépendra de la tâche à réaliser autant que de vos compétences techniques.

En pratique, nous verrons quelques cas d'utilisation d'AppleScript et des scripts Shell pour aller un peu plus loin avec [Automator](#).

Qu'est-ce qu'un script ?

Nous l'avons vu, un script décrit un enchaînement d'*actions* réalisées par une ou plusieurs applications.

Si cela décrit plutôt bien l'idée générale d'un script de traitement, on peut y ajouter beaucoup de subtilités. Ainsi, un script peut intégrer des répétitions ou utiliser un autre script. Comme les chansons ont leur refrain qui se répète, les

scripts peuvent intégrer des structures de contrôle pour décrire les répétitions (les boucles) ou les fragments réutilisables (les fonctions ou sous-procédures).

Un script est exécuté par un logiciel particulier, l'interpréteur de script. En général, un interpréteur n'a pas d'interface graphique. C'est le moteur sous le capot.

Les scripts peuvent être écrits dans une multitude de langages différents. Ce livre ne traite que d'un langage particulier, un langage visuel : les processus Automator.

Sous la carrosserie du robot : comme pour les autres langages de scripts, Automator possède aussi un interprète. Mais vous n'aurez pas affaire à lui, l'éditeur [Automator](#) est la seule application qu'il vous est nécessaire de manipuler. Mac OS X s'occupe pour vous d'appeler le *moteur Automator* pour exécuter votre processus.

La notion de processus Automator

Automator ne parle pas de scripts, mais de *processus*.

L'éditeur [Automator](#) va vous permettre d'assembler des actions qui vont s'exécuter les unes après les autres. Un enchaînement séquentiel qui sera votre processus.

Chacune des actions est une petite usine à traitement. Alimentée par des objets, elle va les transformer ou les utiliser pour en créer de nouveaux. [L'illustration 1.1](#) montre comment une action ingère des éléments en entrée pour les traiter et redonner d'autres éléments en sortie.

Le résultat de chaque action est utilisé pour alimenter l'action suivante.



ILLUSTRATION 1.1 Fonctionnement d'une action

Les éléments reçus alimentent l'action qui produit d'autres éléments

Vous pouvez ainsi cumuler un ensemble de traitements pour obtenir le résultat que vous souhaitez. Regardez l'animation de l'[Illustration 1.2](#) pour voir le principe de fonctionnement d'un processus.

Le caractère très linéaire des enchaînements fait que votre processus reste très simple. Cela ne veut pas pour autant dire simpliste.

Il existe des solutions pour casser un peu ce déroulement linéaire. Nous les étudierons ultérieurement.



ILLUSTRATION 1.2 PROCESSUS
Illustration du fonctionnement d'un processus qui enchaîne trois actions.

Différences entre scripts et processus

Un langage de scripts est comme tout autre langage de programmation : un moyen générique de mettre en œuvre des logiciels plus ou moins complexes. Il possède son vocabulaire et une grammaire propre. Mais en règle générale, un langage de scripts permet de réaliser des petits logiciels sans dépendre d'applications extérieures.

De ce point de vue, [Automator](#) est assez éloigné des modèles de programmation traditionnels. En effet, il n'a pas de vocabulaire propre. À l'exception de quelques éléments de base comme les actions de manipulation de variables, [Automator](#) ne sait pas faire grand-chose.

L'essentiel du vocabulaire d'actions d'[Automator](#) est fourni par les applications de votre Mac. Ces actions sont masquées dans le dossier de chaque application. Vous pouvez voir les actions proposées par OmniOutliner dans l'[Image 1.2](#).

C'est une caractéristique particulière qui le distingue immédiatement de tous les autres

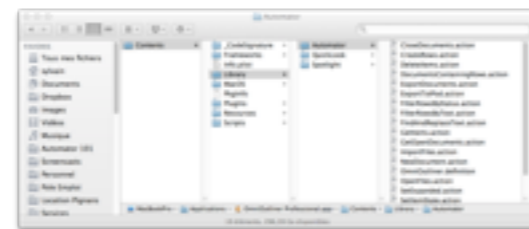


IMAGE 1.2 Actions dans une application
Les actions dans l'application OmniOutliner

langages de scripts.

Pouvoir construire visuellement vos processus est également une distinction de taille, les autres langages adoptant plutôt une forme textuelle.



Oui, sur OS X les applications que vous pouvez voir dans le dossier **Applications**, ou **Utilitaires**, sont en réalité des dossiers qui regroupent, sous la forme d'un paquet tout en ensemble de briques comme l'exécutable de l'application, les ressources graphiques ou multimédias de l'interface graphique, des extensions du système comme les actions pour le système Automator.

Échanges entre applications

La création d'un processus [Automator](#) dépend donc pour l'essentiel des capacités qu'une application propose pour « discuter » avec elle.

Les processus sont un moyen pour piloter une application, mais aussi de les faire dialoguer. Les applications n'ont pas besoin de se connaître directement. Il suffit qu'elles adoptent le vocabulaire d'[Automator](#). Ce dernier se pose ainsi comme médiateur pour intégrer les fonctionnalités de vos applications et en repousser les limites.

Vous pourrez, par exemple, utiliser la synthèse vocale pour énoncer un texte, le transformer en fichier son, puis utiliser Mail pour envoyer un courriel avec le fichier joint.

Vocabulaire d'une application

Chaque application peut être manipulée de différentes façons :

- l'interface utilisateur est le moyen le plus visible : menus, icônes, boutons ;
- les actions que l'application met à disposition d'[Automator](#) pour écrire des processus ;
- le vocabulaire destiné à AppleScript, indispensable de le connaître si vous voulez utiliser ce langage de scripts dans vos processus [Automator](#).

De plus, le système est capable de remplacer les actions de l'utilisateur par un script qui pilotera l'interface à votre place. Cette fonctionnalité permet, par exemple, à Automator d'enregistrer vos actions pour les intégrer dans une action d'un processus.

Même si AppleScript sort du périmètre de ce livre, nous en verrons donc quelques éléments qui peuvent faciliter l'écriture de processus.

Actions proposées par une application

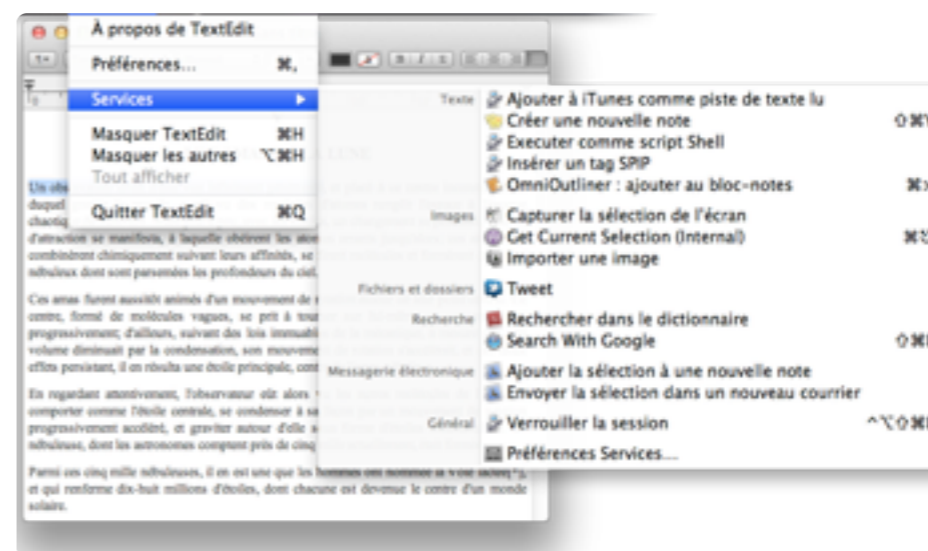
Lorsqu'un éditeur écrit un logiciel, il peut l'ouvrir au monde extérieur en le rendant « *pilotable* ». En ce qui nous concerne, nous n'étudierons que les actions, mais gardez à l'esprit qu'adopter un vocabulaire AppleScript est aussi une possibilité.

Ce n'est pas Automator qui décide de ce qu'une application sait faire ou non. Tout dépend de ce que fournit celle-ci. Lorsque vous installez une application dans votre dossier [Applications](#), le système en fouille le contenu pour y chercher les actions proposées, comme vous pouvez le voir dans l'[Image 1.2](#).

Toutes les actions sont ainsi identifiées par le système et vous pouvez les retrouver dans l'éditeur d'Automator.

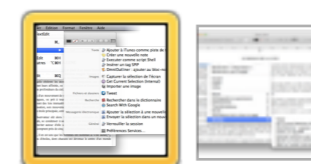
Services offerts par une application

OS X offre un autre moyen d'interaction entre applications : les *services*.



GALERIE 1.1 Le menu des services

Le menu des services tel qu'on peut y accéder à travers le menu de l'application active, ici [TextEdit](#).



Pour résumer, les services sont un moyen simple pour une application de mettre certaines compétences au service d'autres applications. L'appel à un service est fait par l'utilisateur en utilisant le menu **Services** du menu de l'application.

Certaines applications sont aussi capables d'intégrer les services dans les menus contextuels. Nous verrons, par exemple, comment ajouter des services dans le menu des actions proposées par le **Finder**.

Attention. Toutes les applications ne savent pas exploiter les services mis à disposition par les autres applications. Cette fonctionnalité est réservée aux logiciels construits sur le socle de *Cocoa*⁽²⁾. La [Galerie 1.1](#) montre comment **TextEdit** intègre les services d'autres applications.

Automator est le moyen le plus simple et le plus rapide pour construire vos propres services et les faire connaître du système.

Notes :

1. Une image utilisée par Steve Jobs pour parler de l'ordinateur personnel en général et du Mac en particulier.
<http://www.folklore.org/StoryView.py?project=Macintosh&story=Bicycle.txt>.
2. Avant d'être Mac OS X, le système de votre Mac s'appelait NeXT Step. C'est lui qui a introduit la notion de services. Cocoa est le nouveau nom des outils de développement de NeXT Step et seules les applications qui l'utilisent en profitent.



L'application Automator

Ce chapitre est une visite guidée de l'éditeur de processus. Découvrons son interface et ses principales possibilités.



Le fonctionnement d'Automator

Les Actions dans Automator

[Automator](#) vous donne accès à une bibliothèque d'actions. Chaque action qui y est répertoriée vient d'une application spécifique et vous donne accès à une fonctionnalité bien particulière.

Par exemple, l'application [Aperçu](#) vous propose toute une liste d'actions capables de manipuler des images et des documents au format PDF. Mais chaque action reste simple.

La richesse de fonctionnalités des processus dépendra uniquement de vous et de la façon dont vous arrangerez les actions.

Même si l'action que vous utilisez ne propose qu'une fonctionnalité unique, son comportement est pour sa part tout à fait modulable. Pour cela, l'action dispose d'un panneau de contrôle. C'est à travers ce dernier que vous allez définir les valeurs de ces paramètres. Ainsi, si une action vous permet de copier un fichier, l'un de ses paramètres vous permet d'indiquer dans quel dossier le nouveau fichier devra être créé.

Tous les objets modifiés ou construits par votre action vont servir de sortie. Si votre action est la dernière, cette sortie va pouvoir alimenter le contexte d'appel. Si votre action est encore suivie d'une autre, ce résultat sera transmis à celle-ci pour lui servir d'entrée.

Fournir des données aux actions

Chaque action est ainsi alimentée par des objets. Ces objets viennent pour l'essentiel, et comme nous l'avons déjà vu, de l'action précédente dans la séquence du processus.

Mais il existe d'autres moyens d'alimenter une action en objets :

- le contexte d'utilisation du processus ;
- l'enchaînement en cascade des actions au fur et à mesure de l'exécution du processus ;
- l'interaction avec l'utilisateur ;
- les variables.

Le contexte de l'application à partir de laquelle le processus est utilisé peut aussi servir de données d'entrée pour sa première action. Par exemple :


- vous pouvez lancer un processus en glissant des documents sur son icône ;
- vous pouvez lancer un processus dans un éditeur de texte en lui envoyant le texte sélectionné dans cet éditeur ;
- vous pouvez aussi avoir un processus qui n'utilise aucune donnée d'entrée, mais qui va aller chercher les données à traiter directement dans une application.

Dans ce dernier cas, vous devrez utiliser des actions d'extraction de données fournies par les logiciels. Le carnet d'adresses peut ainsi vous permettre de récupérer tout ou partie de vos contacts. L'agenda sait faire de même avec vos tâches et vos rendez-vous.

Dans le second cas, c'est l'enchaînement naturel des actions dans le processus qui alimente chaque action avec le résultat de la précédente.

Il existe une troisième alternative dans laquelle votre processus dialogue avec l'utilisateur pour obtenir des informations. Ce dialogue peut être mis en place de deux façons différentes :

- Vous utilisez une action d'interrogation. Elle va interrompre la séquence automatique pour que l'utilisateur fournisse les informations nécessaires.
- Vous pouvez également vous appuyer sur [Automator](#) pour présenter à l'utilisateur le panneau de contrôle d'une action. L'utilisateur a alors la possibilité de voir les choix faits par l'auteur du processus, mais il garde la main et peut éventuellement modifier ces choix par des valeurs plus appropriées au contexte.

 Un processus est normalement fait pour vous faciliter la tâche et rendre les choses les plus automatiques possible. Si votre script impose de faire trop de choix à l'utilisateur, vous aurez le comportement d'un assistant. Ce choix d'ergonomie est le vôtre et dépend largement de l'audience à laquelle votre processus est destiné.

Le dernier moyen d'alimenter une action est de lui fournir les valeurs de variables. C'est exactement ce que nous allons détailler dans le paragraphe suivant.

Les variables

Pour ce que nous avons vu des processus [Automator](#), ils se comportent comme une chaîne d'assemblage en ligne droite. En réalité, cette ligne droite possède quelques chemins de traverse. Vous pouvez à tout moment stocker la valeur des objets dans des casiers de rangement, sur le côté de votre chaîne.

[Automator](#) ne parle pas de casiers, mais de variables.

Nous verrons en détail comment ajouter des variables dans les processus, et comment en utiliser le contenu. Le principe général est d'utiliser des actions particulières.

Pour vous faciliter la vie, le moteur [Automator](#) met quelques valeurs communes à votre disposition dans des variables. Vous y trouverez par exemple la

date du jour et le nom de l'utilisateur. Ces variables ne peuvent pas être modifiées. Il n'est possible que d'en lire la valeur.

Les variables sont utilisables pour alimenter une action et, pour certaines d'entre elles, donner une valeur à un paramètre de fonctionnement de l'action.

Interface graphique

Lancer Automator

Maintenant que nous avons fait le tour des idées qui sous-tendent Automator, il est grand temps de commencer à découvrir cette application par vous-même.

La première chose à faire est de lancer l'application. Si vous êtes comme moi, vous pouvez simplement utiliser **Spotlight** pour rechercher et lancer l'application (chercher « Autom » devrait suffire).

Si vous n'utilisez pas Spotlight, il vous reste le **Launchpad** disponible depuis Mac OS X 10.7 Lion. Dans le **Finder**, vous trouverez **Automator** dans le dossier **Applications**.

Une fois l'application lancée, vous devriez avoir un écran d'accueil ([Interactif 2.1](#)) montrant l'interface de l'application et une feuille de démarrage pour choisir le type de processus à créer.

Pour l'instant, nous allons simplement explorer les éléments de l'interface. Choisissez simplement l'icône **Processus**, puis validez en cliquant sur le bouton **Choisir**.

La feuille d'accueil va s'effacer pour vous laisser découvrir toute l'interface d'**Automator**.



INTERACTIF 2.1 L'écran d'accueil d'**Automator**

Structure de l'interface

L'écran de l'éditeur d'[Automator](#) est composé de trois parties que vous pouvez explorer dans le cadre [Interactif 2.2](#) :

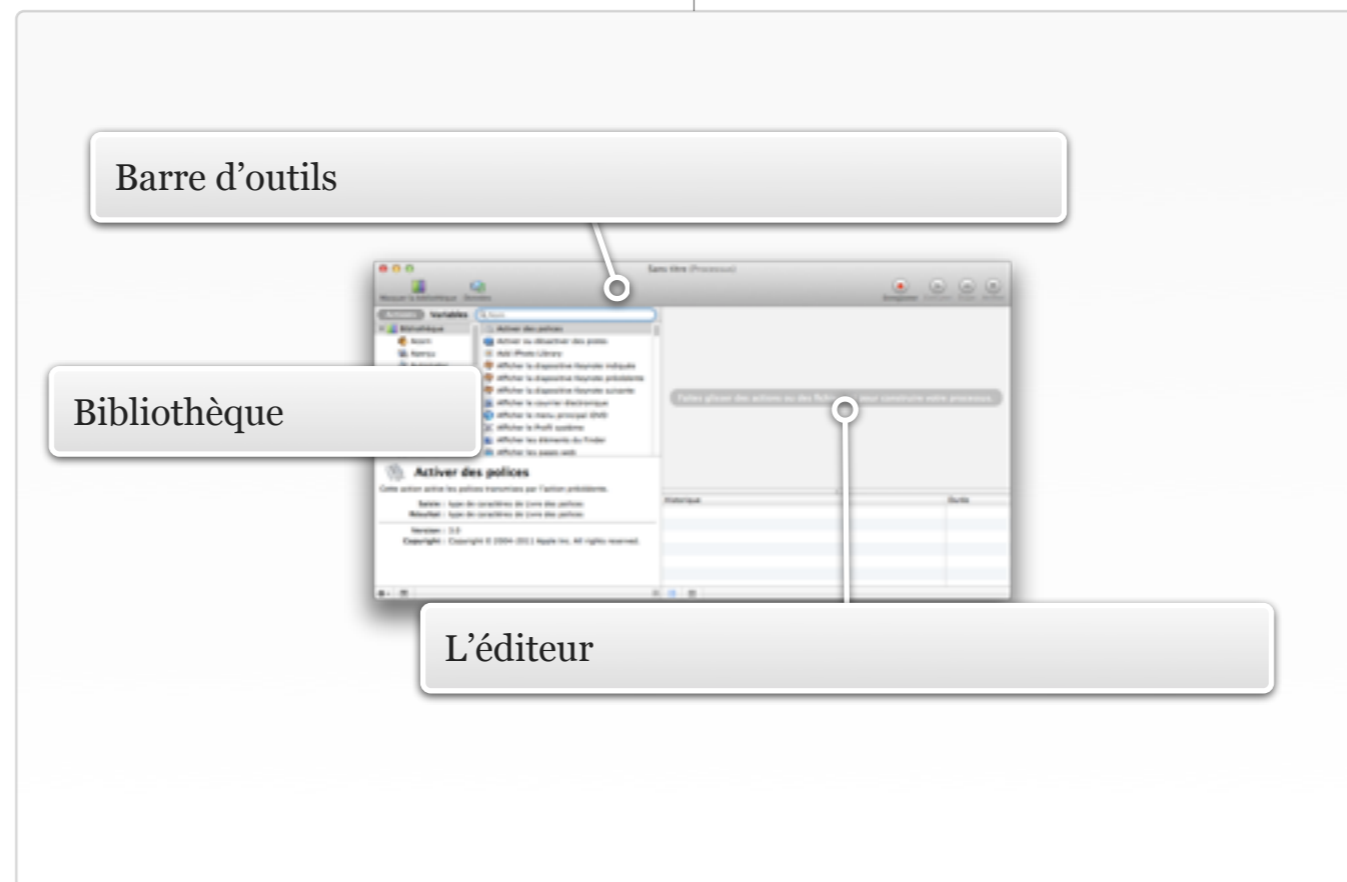
1. une barre d'outils ;
2. la bibliothèque d'actions et de variables ;
3. l'éditeur.

! En haut de la fenêtre, la barre d'outils est habituelle sur les applications Mac. Elle est facilement personnalisable à travers un menu de personnalisation qui s'obtient d'un clic avec le bouton droit de la souris, ou avec la combinaison **Ctrl + clic**.

Pour plus de clarté, je vous demande de bien vouloir conserver l'affichage du texte des boutons dans la barre d'outils. Cela vous permettra de retrouver rapidement les boutons dont je parle par la suite.

Verticalement la fenêtre est séparée en deux parties :

1. à gauche : la bibliothèque d'actions ;
2. à droite : l'éditeur de processus proprement dit.

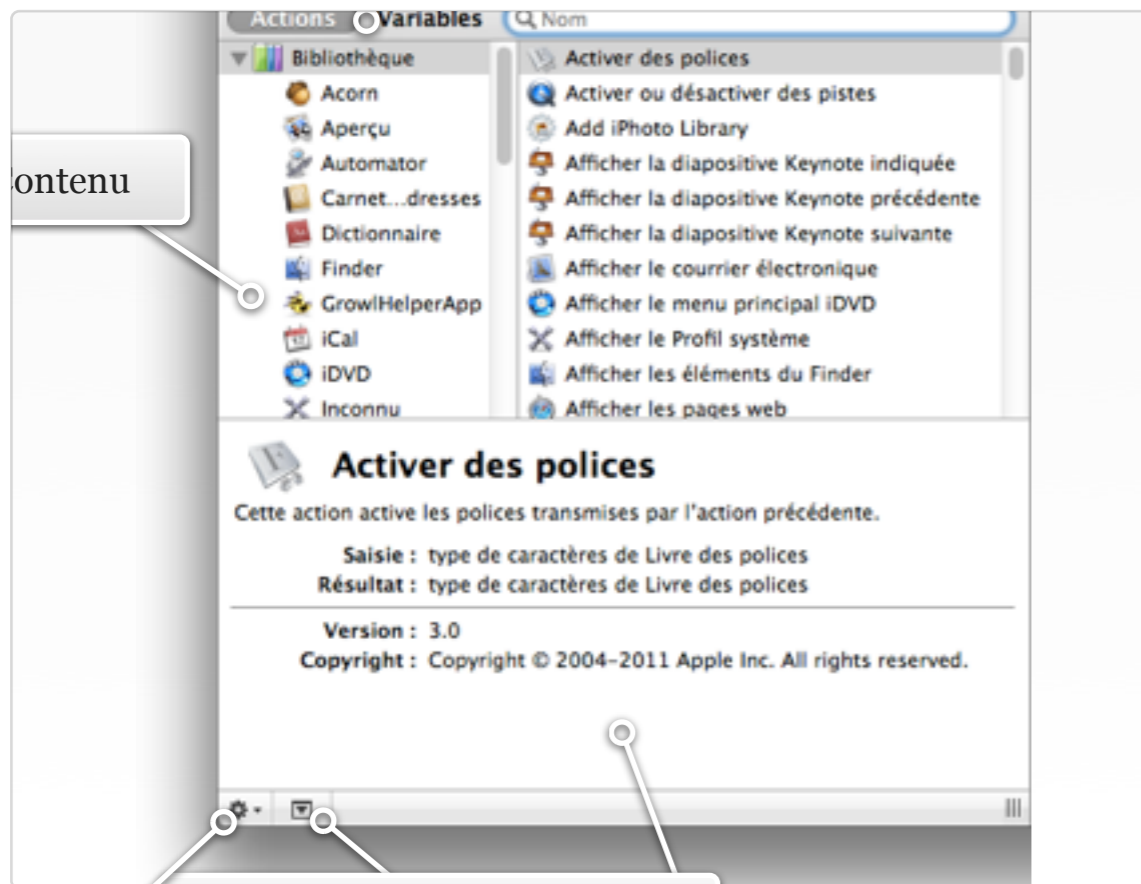


INTERACTIF 2.2 La fenêtre d'[Automator](#)

La bibliothèque

La bibliothèque se compose de cinq parties visibles dans le cadre [Interactif 2.3](#) :

1. En haut à gauche, on peut basculer entre la bibliothèque des actions et celle des variables ;
2. En haut à droite, un champ de recherche permet de filtrer les actions ou les variables affichées ;
3. Le contenu de la bibliothèque s'affiche dans deux colonnes :



INTERACTIF 2.3 Les éléments de la bibliothèque

- à gauche une vue hiérarchique présente les différents éléments de la bibliothèque ainsi que des groupes automatiques ou manuels ;
 - à droite la liste des actions, ou variables, correspondant à la section sélectionnée dans la colonne de gauche.
4. Sous le contenu de la bibliothèque, on trouve une documentation de l'action, ou de la variable, sélectionnée.
 5. Enfin, tout en bas, une petite barre de boutons permet de construire vos groupes d'actions et de variables ou de masquer, et montrer, la documentation de la sélection.

Lorsqu'une action est sélectionnée, sa documentation est affichée comme dans la [Image 2.1](#).

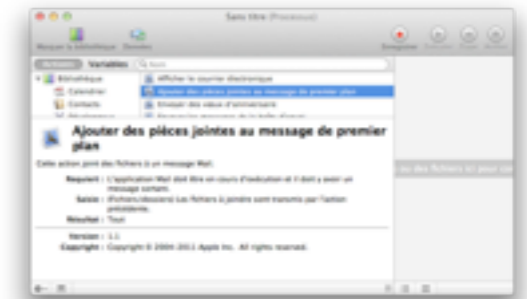


IMAGE 2.1 Documentation d'une action dans la bibliothèque

Chaque action porte sa fiche de capacité :

- Que fait l'action ?
- Quels sont les objets acceptés en entrée ?
- Quel est le résultat de l'action ?
- Quelle est l'interface graphique pour personnaliser l'action ?
- Quelle est la documentation de l'action ?

Organisez votre bibliothèque

Comme vous pouvez le voir, cette première partie vous permet d'explorer tous les détails de la bibliothèque des actions disponibles. Vous avez également toute latitude pour personnaliser cet affichage en construisant vos propres regroupements d'actions en fonction de vos préférences et de vos besoins.

Par exemple, nous pouvons très facilement construire un « *groupe intelligent* » pour rapidement identifier les actions qui manipulent des documents PDF.

1. Cliquez sur le bouton en forme de roue dentée et sélectionnez **Nouveau groupe intelligent**.
2. Une feuille glisse du haut de votre fenêtre pour définir le nom de votre groupe et les critères qui en définissent le contenu.
3. Donnez un nom comme « traitement sur PDF ».
4. N'indiquez qu'une seule règle : le nom contient PDF.
5. Validez par le bouton **OK**.

Et voilà, votre bibliothèque devrait maintenant contenir un nouveau groupe référant les actions dont le nom fait référence au format PDF.

Les règles peuvent regrouper plusieurs critères et comme votre groupe est « intelligent » il se mettra à jour tout seul au fur et à mesure que des applications ajouteront des actions à votre système.

Explorez les différentes actions et variables connues de votre Mac. Faites-vous une petite idée des choix qui sont à votre disposition pour construire des processus.

! La bibliothèque peut être masquée et affichée avec un bouton de la barre d'outils. Une entrée du menu **Présentation** vous permet également de gérer cet affichage.

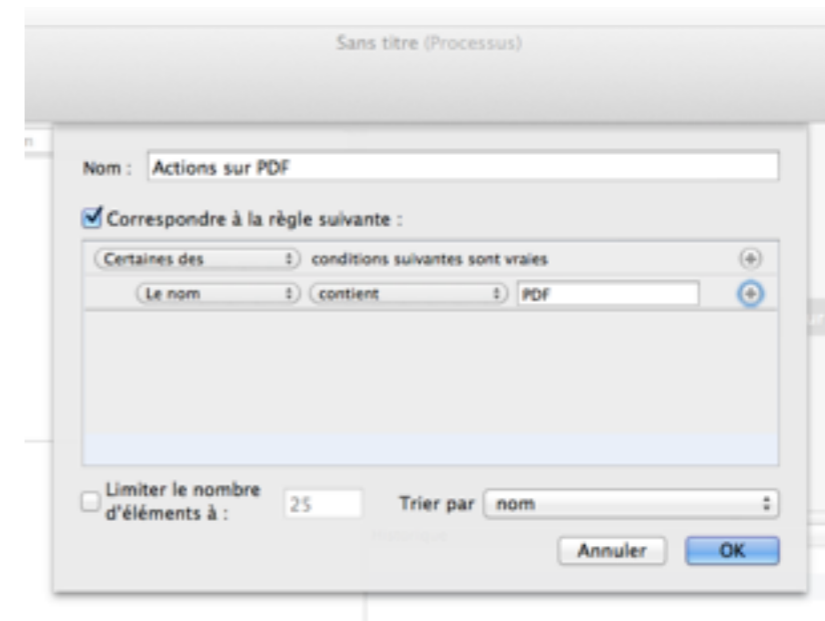
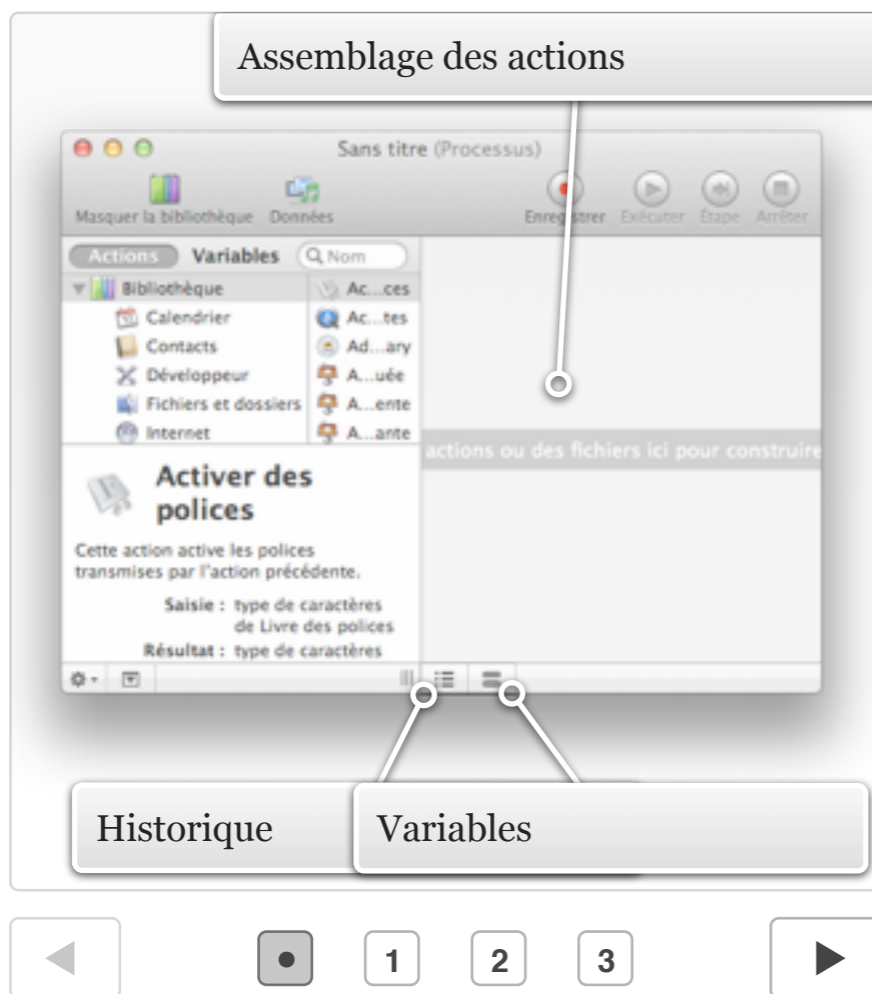


IMAGE 2.2 Construire une liste intelligente

Éditeur de processus

L'éditeur de processus est certainement la partie la plus importante de la fenêtre Automator. C'est là que vous allez devoir construire, assembler et ordonner les actions de votre processus ([Interactif 2.4](#)).



INTERACTIF 2.4 Les éléments de l'éditeur

Les actions sont ajoutées en les faisant glisser de la bibliothèque vers l'éditeur. Elles sont ajoutées lorsque vous relâchez le bouton de la souris au meilleur endroit possible.

Ainsi, si vous déposez une action entre deux actions déjà présentes dans le processus, ces deux actions s'écartent pour laisser la place à votre nouvelle action, comme dans l'[Image 2.3](#).

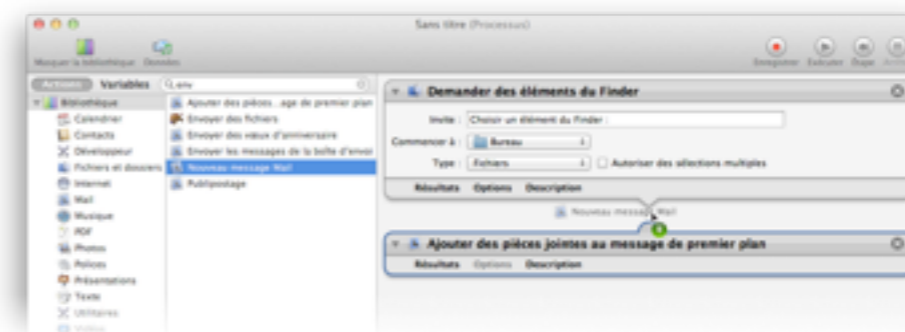


IMAGE 2.3 Insertion d'une action

La même méthode est utilisable pour réordonner les actions. Vous cliquez, en maintenant le bouton pressé, sur une action, puis vous la faites glisser à une nouvelle position dans la séquence. Le déplacement de l'action décalera les actions existantes, indiquant la nouvelle position possible, comme cela est illustré dans l'[Image 2.4](#).

Et voilà, vous avez maintenant les connaissances essentielles pour construire une séquence d'actions.

Comme dans la partie de la bibliothèque, vous aurez certainement remarqué que le bas de l'éditeur contient un bandeau de petites icônes ([Interactif 2.4](#)). Ces icônes contrôlent l'affichage de deux outils importants.

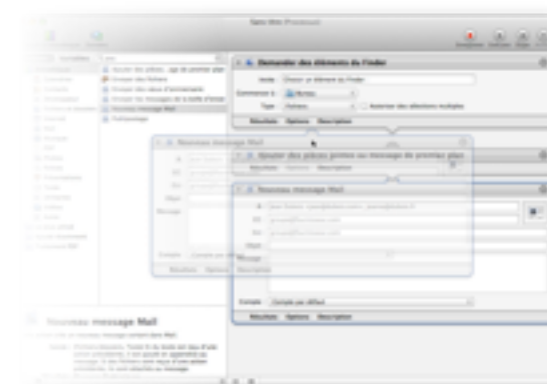


IMAGE 2.4 Déplacement d'une action

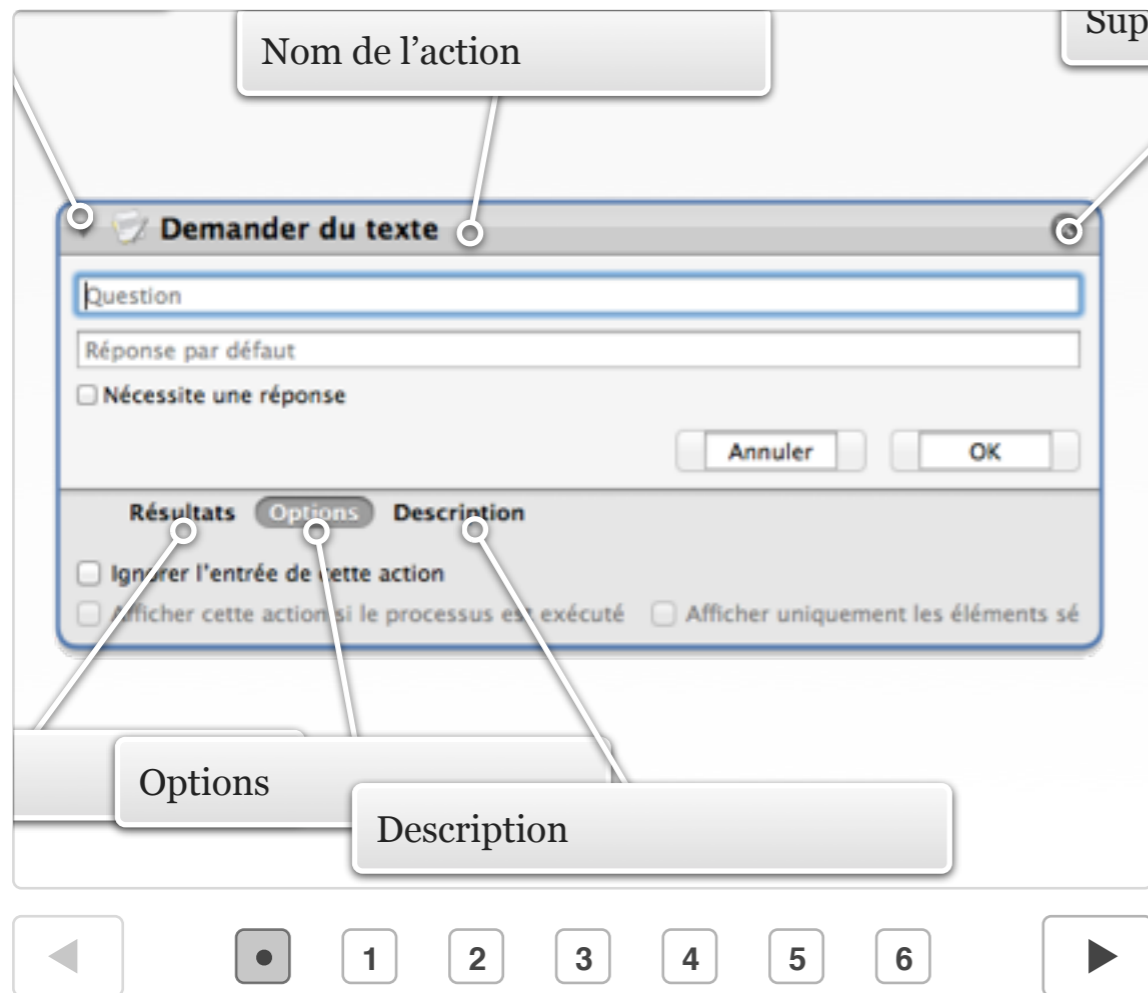
Le premier bouton permet d'afficher ou de masquer l'historique des actions. Lorsque vous allez exécuter un processus à partir d'Automator, vous pourrez ainsi vérifier comment se déroule chaque étape, et même en combien de temps elle s'exécute.

Le second bouton contrôle l'affichage, ou le masquage, des variables de votre processus. Pour l'instant, cette liste est vide, car vous n'avez pas de script et donc aucune variable personnelle.

Anatomie d'une action

Toutes les actions se présentent de la même façon : un cadre dont le titre contient le nom de l'action.

Ce cadre contient plusieurs boutons qui contrôlent l'affichage et le comportement de l'action, comme vous pouvez le voir dans le cadre [Interactif 2.5](#) :



INTERACTIF 2.5 Représentation d'une action

- Un petit triangle permet de plier ou déplier le bloc de l'action. Plier les actions améliore la lisibilité d'un processus qui en comprend de nombreuses.
- Le cercle contenant une croix permet de supprimer une action du processus. Cette action ne demande aucune confirmation, mais la

suppression peut facilement être annulée avec le menu **Édition > Annuler Supprimer l'action**.

- Le contenu de la boîte d'action dépend uniquement de l'action que vous aurez choisie. Cet élément d'interface est directement fourni par l'action.
- Le bas de la boîte contient une série de boutons qui affichent, ou masquent, des informations et options complémentaires.

Ces trois éléments sont les suivants :

- **Résultats** est utilisé pour tester vos processus dans [Automator](#). Il affiche les objets retournés par l'action lors de la dernière exécution du processus à partir de l'éditeur de processus. Un bon moyen pour vérifier que votre processus fait bien ce que vous attendez de lui. Les options de présentation des **Résultats** permettent de choisir entre icônes, liste et une représentation textuelle du résultat.
- **Options** permet de contrôler le comportement de l'action lors de l'exécution du processus. Sans entrer dans le détail immédiatement, c'est ici que vous pouvez demander à l'action de présenter ses paramètres à l'utilisateur pour rendre votre processus un peu plus automatique.
- **Description**, enfin, vous permet de visualiser la documentation de l'action sans devoir retourner la chercher dans la bibliothèque. Un moyen pratique pour gagner du temps si vous n'êtes pas sûr de son comportement.

Échanges entre applications

Nous n'allons pas dès maintenant construire un processus. Mais il est important de concrétiser cette première découverte de l'éditeur d'Automator.

Un processus minimum

En utilisant votre connaissance toute fraîche de l'utilisation de la bibliothèque, cherchez-y une action **Demander du texte**.

Faites maintenant glisser cette action dans l'éditeur. Votre curseur devrait s'enrichir d'une bille verte contenant un signe plus (« + »). Relâchez le bouton de la souris au-dessus de l'éditeur vide.



Vous venez de réaliser un premier processus qui ne contient qu'une seule action.

Personnalisons un peu cette action en indiquant les informations suivantes :

- Saisissez « Que dois-je dire ? » dans le champ indiquant la question.
- Saisissez « Bonjour, tout le monde ! » dans le champ de la réponse par défaut.

Vous pouvez maintenant exécuter ce premier processus en cliquant sur le bouton **Exécuter** de la barre d'outils.

Vérifiez maintenant que l'historique a bien été mis à jour par le déroulement du processus. Vous pouvez aussi regarder le résultat de l'action.

Une action plus loin

Ajoutez maintenant une nouvelle action à la fin de votre processus. Disons que vous choisissez l'action **Énoncer le texte**. Exécutez le processus à nouveau. Si le son de votre ordinateur n'est pas coupé, vous devriez entendre une voix de synthèse prononcer le texte que vous avez saisi au clavier dans le dialogue.

Bravo, vous avez réussi un premier processus. Pas très utile, mais loquace.



Ce qui est le plus intéressant c'est que la forme des boîtes d'action a certainement été modifiée comme vous pouvez le voir dans l'[Image 2.5](#).

1. Le bas de la première action doit maintenant avoir la forme d'une petite flèche dirigée vers le bas.
2. Le haut de la seconde et dernière action doit présenter une petite bulle.

La flèche indique que le résultat de l'action est transmis à l'action suivante qui le reçoit dans cette connexion en forme de demi-cercle.

C'est par ce lien visuel que vous savez que l'action **Énoncer le texte** va prononcer le texte renvoyé en sortie de l'action précédente **Demander du texte**.

Voilà, vous savez décrypter la boîte d'une action et identifier les échanges entre ces actions. Nous voilà prêts à mettre en œuvre notre premier véritable processus.

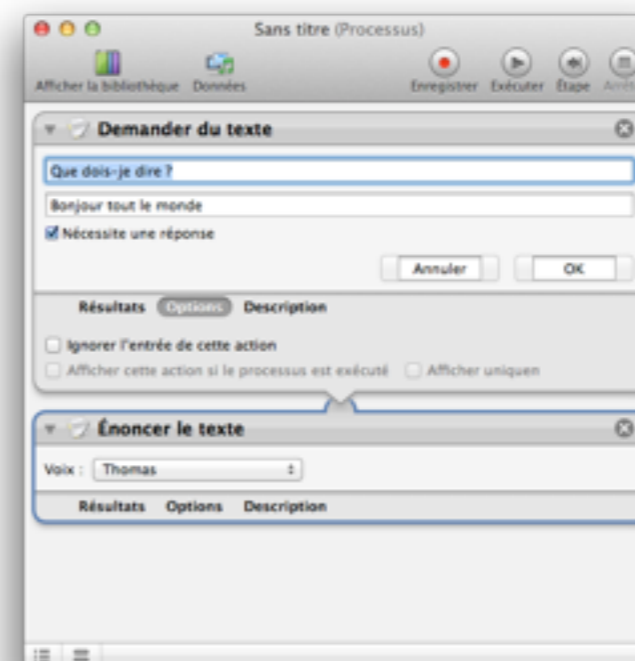


IMAGE 2.5 Connexion de deux actions

Mise en pratique

Afin de vous mettre en perspective les paragraphes précédents, nous allons construire un processus très simple, mais plus complet que l'exemple précédent, pour archiver des fichiers puis les supprimer.

Ce contenu de ce processus est illustré dans l'[Image 2.6](#).

La vidéo [Vidéo 2.1](#) illustre, dans le détail, comment construire ce processus.

Assembler

La première étape est bien entendu de construire votre assemblage d'actions en utilisant l'éditeur. Cela peut être réalisé de différentes façons :

- glissé-déposé des actions de la bibliothèque vers la zone d'édition ;
- double-clic sur une action ;
- avec le clavier : déplacez-vous dans la liste d'actions et ajoutez-en une en validant avec la touche **Entrée**.

Ordonner

Dans un second temps, vous pouvez vouloir réorganiser votre processus en réarrangeant ses actions :

- le glissé-déposé des actions reste la façon la plus intuitive ;

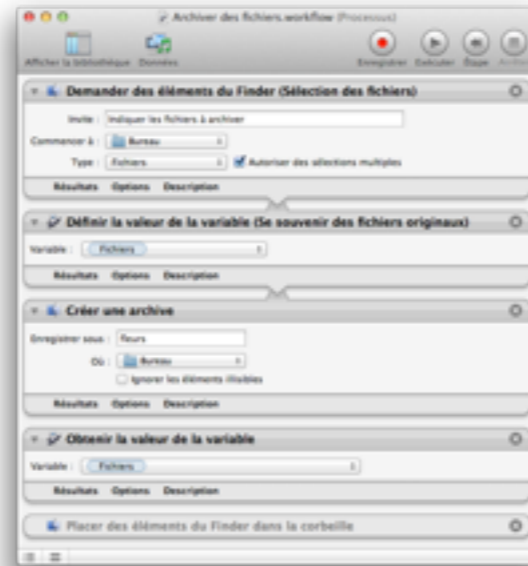


IMAGE 2.6 Exemple pour archiver des fichiers

- avec le clavier, vous pouvez utiliser la combinaison de touches **⌘ + haut** et **⌘ + bas** pour déplacer les actions sélectionnées dans l'éditeur.

Documenter

Pour rendre vos processus plus lisibles, il est également possible de donner un nom à chaque action de votre processus. Ceci n'est permis que par l'intermédiaire du menu contextuel attaché à chaque action : **clic-droit** (ou **Ctrl + clic**).

Désactiver

Lors de la phase d'écriture et de tests, il peut aussi être utile de désactiver une action destructrice, comme la suppression d'un fichier. Ceci est également possible avec le menu contextuel des actions.

Par exemple, lors de la phase de tests, il peut être utile de désactiver une action qui serait destructrice, comme la suppression de fichiers.



VIDÉO 2.1 Exemple de création de processus

Les processus

Dans ce chapitre, nous verrons quels sont les types de processus que permet de créer [Automator](#).

Vous y verrez également quelques exemples simples.



Les types de processus

Avant de poursuivre, il est temps de faire un petit tour d'horizon des différents types de processus qu'[Automator](#) met à votre disposition. C'est l'objet de cette première section.

Processus

Présentation

Les processus simples ne peuvent pas être exécutés en dehors de l'application [Automator](#).

Autre limitation, ces processus ne sont pas capables de recevoir des objets du contexte d'exécution. Impossible, par exemple, de faire glisser une icône de document sur celui du processus pour lui appliquer un traitement.

Un processus créé avec ce type sera par défaut sauvé comme processus. Rien ne vous empêche de modifier alors le format en application. Plus de détails sont donnés dans le paragraphe suivant.

Les processus seront surtout utiles le temps de faire un prototype. Mais dès que vous estimerez ce dernier prêt, vous voudrez certainement l'enregistrer en utilisant le format application.

En action

Visionnez-la [Vidéo 3.1](#) qui montre comment exécuter un processus simple.



IMAGE 3.1 Un processus simple, sans données d'entrée



VIDÉO 3.1 Exécuter un processus simple avec Automator

Les actions

Ce chapitre est une visite guidée de quelques actions choisies.

Illustrés d'exemples d'utilisation, j'espère que cela vous aidera à trouver l'inspiration pour vos propres processus.



Introduction

Dans ce chapitre, je vous propose une petite visite guidée des principales actions disponibles dans [Automator](#). Cet inventaire n'a rien d'exhaustif et son but n'est que de vous mettre le pied à l'étrier. Pour correctement maîtriser l'outil, vous devrez certainement parcourir vous-même la bibliothèque d'actions et chercher sur le web d'autres actions.

N'oubliez pas que bon nombre d'applications Mac proposent leurs propres actions. Il peut donc même être utile, selon vos besoins, de choisir vos applications en fonction des capacités d'automatisation qu'elles proposent.

Aperçu

Les actions regroupées sous la bannière d'[Aperçu](#) sont principalement dédiées au traitement des images et des documents PDF. Nous allons ici étudier quatre actions qui manipulent toutes des images. Une section particulière sera dédiée aux traitements des documents PDF.

Ajouter une bordure aux images

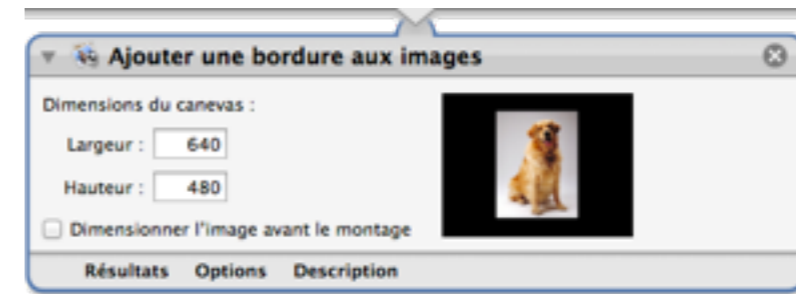


IMAGE 4.1 L'action **Ajouter une bordure aux images**

- 🎯 Important : Cette action modifie les fichiers qu'elle obtient en entrée. C'est pourquoi lorsque vous l'ajoutez à un processus Automator vous propose de la précéder d'une copie des fichiers.

Le nom de l'action est quelque peu trompeur. Certes, il est possible d'avoir une bordure, noire, autour des images transmises, mais à la condition que la taille indiquée à l'action soit supérieure à la taille originale de l'image ! Si la taille que vous indiquez est inférieure à la taille originale, non seulement vous n'aurez pas une bordure, mais le résultat sera une image simplement recadrée.

En réalité, cette action a pour effet de modifier la taille du canevas d'une image, la taille de cette dernière pouvant être adaptée ou conservée.

Si vous cochez l'option **dimensionner l'image avant le montage** votre image sera correctement rétrécie ou élargie pour rentrer dans les dimensions indiquées ([Image 4.1](#)). Au fait, largeur et longueur sont indiquées en pixels. Pas question d'utiliser des unités métriques. Il n'est pas possible d'utiliser une valeur de variable pour définir la nouvelle taille du canevas de l'image.



IMAGE 4.2 La photo originale (1 600 x 1 200 pixels)

Je vous conseille fortement

d'utiliser cette action qu'à la condition d'avoir au préalable obtenu la taille de l'image originale, voire redimensionné celle-ci. Au pire, vous pouvez afficher l'action en cours pour que l'utilisateur saisisse lui-même les dimensions.

Pour illustrer ce comportement, j'ai pris une photo de 1 600 x 1 200 pixels que je vais recadrer avec différents paramètres ([Image 4.2](#)).

Nous allons construire un processus application simple comprenant trois actions :

1. La copie du fichier original sur le bureau.
2. Le changement du nom de la copie.
3. L'ajout de bordure.

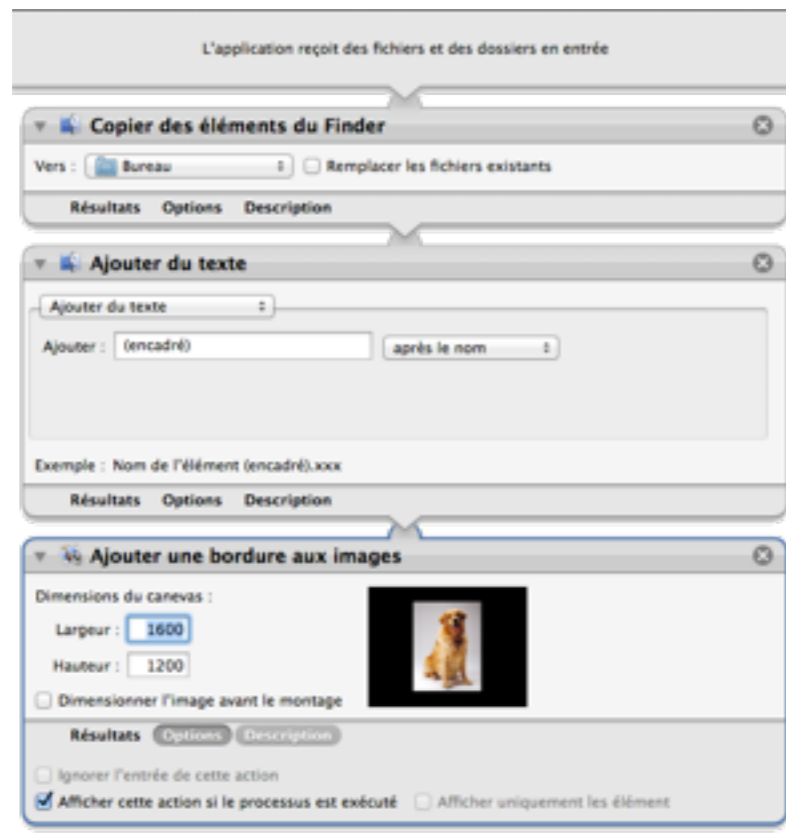


IMAGE 4.3 Application illustrant **Ajouter une bordure aux images**

Il vous suffit maintenant de glisser une image sur cette application pour que le processus se déclenche. Pensez à cocher l'option afficher cette action si le processus est exécuté, cela vous permettra de jouer avec les options de l'action.

! Cette application n'est pas très robuste. Elle ne peut pas être exécutée deux fois sans que vous effaciez ou renommez le fichier image produit. Trouvez comment arranger cela !

Dans un premier temps, nous allons ajouter une grosse bordure (200 pixels) à cette image. Nous n'allons pas modifier la taille du canevas, sans redimensionner l'image elle-même. Nous avons bien l'ajout d'une bordure noire comme vous pouvez le voir sur l'[Image 4.4](#).



IMAGE 4.4 Ajout de 200 px sur les deux dimensions, sans recadrer l'image

Si maintenant nous donnons une taille plus petite pour le canevas cible, toujours sans redimensionner l'image, nous obtenons une image recadrée. Comme ce recadrage ne crée pas de vide dans le canevas, il n'y a pas de bordure. Le résultat est visible sur l'[Image 4.5](#).



IMAGE 4.5 Canevas réduit à 600 x 200 sans redimensionnement

Pour un dernier exemple, nous allons utiliser le redimensionnement, tout en réduisant la taille du canevas. En revanche, nous allons créer des espaces dans ce dernier qui ne pourront pas être remplis par l'image. Nous aurons donc un fond noir à ces endroits, comme sur l'[Image 4.6](#).

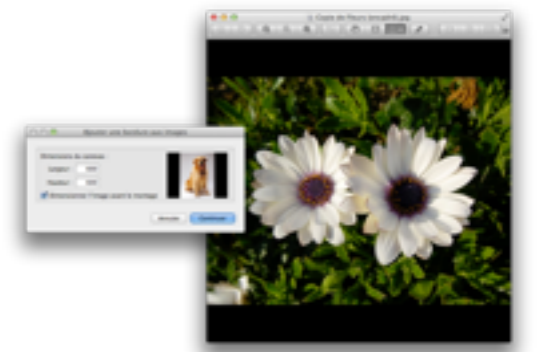


IMAGE 4.6 Canevas plus petit et image redimensionnée

Et voilà en quelques exemples comment changer la taille du canevas d'une image. On peut regretter que l'action n'offre pas un peu plus de souplesse, comme la possibilité de

définir la couleur utilisée pour le cadre. Dommage aussi que la taille actuelle de l'image ne soit pas utilisée par l'action.

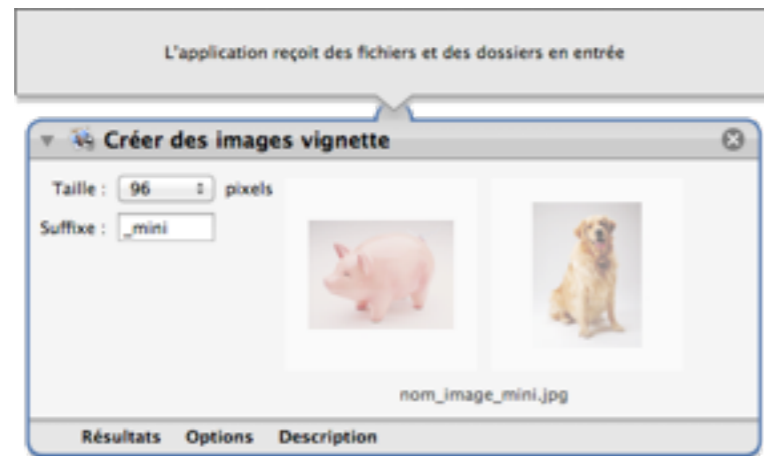


IMAGE 4.7 Une application illustrant l'action **Créer des images vignette**

Vous pouvez cependant trouver des usages pour cette action, par exemple pour créer rapidement des vignettes de photo avec une taille standardisée. De quoi automatiser la création d'une galerie photo dans une page web ou dans un autre type de document.

Les services

Automator brille par l'extrême simplicité qu'il propose pour construire des services dans OS X.

Un moyen qui reste, à mon avis, le plus simple pour utiliser des processus et étendre les capacités d'une application.



La notion de services

Comme nous l'avons vu dans l'introduction de ce livre, les services sont un moyen pour une application d'exporter une fonctionnalité à travers tout le système. On parlera donc d'applications fournisseuses de services. Les applications qui utilisent un service seront donc désignées comme les clientes du service.

Les services sont accessibles par l'intermédiaire du menu du même nom que l'application, mais aussi dans les menus contextuels de l'application cliente.

Nous allons maintenant voir comment [Automator](#) peut être utilisé pour écrire des processus qui vont s'intégrer dans le système comme des fournisseurs de services.

Automator est capable de créer deux types de services :

- les services qui s'appliquent à une sélection de texte ;
- les services qui s'appliquent sur une sélection de fichiers ou dossiers.

Pour une description plus précise des variantes possibles dans chacune de ces catégories reportez-vous à la description des processus de type service dans la section décrivant les types de processus ([chapitre « Les Processus », section 1](#)).

Les spécialistes

Non, ce chapitre n'est pas une épisode de la série de films *Expandable*.

Nous y parlerons de processus, mais de processus qui ciblent des domaines bien précis : impression et actions de dossiers.



Les modules d'impression

Le principe général de fonctionnement ainsi que l'utilisation des modules d'impression ont été décrits au [chapitre 2, section Processus](#). Pour rappel :

- Les modules d'impression sont des processus qui ne manipulent que des documents au format PDF.
- On peut exécuter ces modules à partir du dialogue d'impression, en utilisant le menu accessible via le bouton **PDF**, comme cela est visible sur l'[Image 6.1](#)

Ce chapitre a pour objectif de vous présenter les modules d'impression à travers un exemple simple. Vous allez écrire un processus que vous pourrez utiliser lorsque vous souhaitez enregistrer le contenu d'une page internet de façon sécurisée. Un moyen simple pour conserver un relevé de compte, ou une facture, sans que le contenu en soit lisible.



IMAGE 6.1 Le menu PDF

Le dialogue d'impression propose un menu PDF, que ce soit dans sa version complète, comme ici, ou que ce soit dans sa version compacte.



Les actions de dossier

Déclenchement des processus

Les actions de dossier permettent de déclencher automatiquement un processus lorsqu'un ou plusieurs fichiers sont ajoutés à un dossier en observation.

Le fonctionnement s'appuie sur l'idée de surveillance de dossiers. Il n'est pas possible de surveiller des dossiers de façon implicite. Vous devez indiquer ceux qui le seront. De toute façon, [Automator](#) vous y oblige lorsque vous écrirez votre processus.

Votre processus ne peut pas être déclenché lorsqu'un évènement se produit dans un ensemble de dossiers. Un, et un seul, dossier peut être associé à votre processus. En revanche, vous pouvez créer autant d'actions de dossier que vous le souhaitez.

! les actions de dossier ne s'appliquent qu'au dossier sous surveillance. Elles ne sont en aucun cas déclenchées par les ajouts de fichiers dans les sous-dossiers de ce dossier. La surveillance n'est pas récursive !

Lorsque vous créez un processus de type action de dossier, [Automator](#) va prendre en charge la configuration du système pour déclencher la surveillance et l'activer sur le dossier qui est associé à votre processus. Nous verrons aussi dans ce chapitre comment affiner cette configuration.

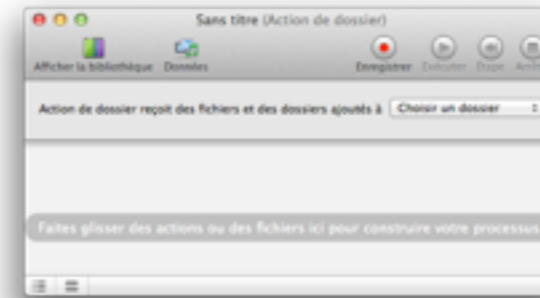


IMAGE 6.2 En-tête d'un processus de type action de dossier

Lorsque vous créez une action de dossier, l'en-tête du processus vous propose de sélectionner le dossier pour lequel l'ajout de fichiers va déclencher le processus, comme vous pouvez le voir dans l'[Image 6.2](#).

Une fois qu'un dossier est associé au processus, et que ce dernier est enregistré, chaque ajout d'un fichier déclenchera l'exécution du processus.

Exemple : publication d'images sur un serveur

Nous allons maintenant voir comment réaliser une action de dossier simple en quelques étapes. Notre objectif est de publier des images automatiquement sur un serveur FTP lorsqu'elles sont ajoutées à un dossier.

Nous allons également mettre en œuvre un peu d'interactivité dans ce processus en introduisant une demande de confirmation pour l'utilisateur. Dans un cas général, cette confirmation n'est pas vraiment utile puisque nous voulons mettre en place un système automatique. Cependant, c'est pour nous un moyen de découvrir comment ajouter un peu d'interactivité dans vos processus tout en modulant le caractère linéaire de leur exécution.

En ce qui concerne la publication vers un serveur FTP, vous pouvez chercher dans la bibliothèque d'actions, mais vous ne trouverez pas d'actions permettant de téléverser un fichier de votre ordinateur vers un serveur distant.

Certes, le [Finder](#) est capable de se connecter sur un serveur de ce type, mais sans pouvoir y écrire, ce qui est malheureusement ce que nous souhaitons.

Important : pour pouvoir réaliser complètement ce processus, vous devez avoir accès à un serveur FTP. Pour cela vous pouvez créer simplement un compte gratuit chez un hébergeur comme [Free^{\(1\)}](#). Avec quelques efforts vous pouvez aussi

utiliser le serveur intégré à Mac OS X Lion ou installer votre propre serveur avec PureFTPd⁽²⁾.

Aller plus loin

« Aucune limite à mon pouvoir ! »

Chancelier PALPATINE,

La revanche des Siths

