

Fichier Lisez-moi Scripts Adobe® InCopy® 2021

Ce document présente des informations sur l'élaboration de scripts dans Adobe InCopy 2021, comme suit :

- ◆ Une synthèse de la documentation relative aux scripts d'InCopy (voir la section "[Documentation relative aux scripts d'InCopy](#)", page 1)
- ◆ Des instructions pour l'exécution d'un script (voir la section "[Exécution de scripts](#)", page 2)
- ◆ Une liste de scripts d'exemple InCopy assortis d'une description succincte pour chacun (voir la section "[Scripts d'exemple](#)", page 2)
- ◆ Une liste des problèmes connus relatifs aux scripts d'InCopy (voir la section "[Problèmes connus liés à l'élaboration de scripts InCopy](#)", page 3)

Pour obtenir plus de détails sur les scripts InCopy, aller à la page de la documentation pour développeur d'InDesign, à l'adresse <https://www.adobe.io/apis/creativecloud/indesign.html>, ou consulter le forum des utilisateurs pour les scripts InCopy (en anglais), à l'adresse http://forums.adobe.com/community/incopy/incopy_scripting.

Pour obtenir des informations de dernière minute sur les scripts InCopy, voir la dernière version en date de ce fichier sur la page de la documentation pour développeur d'InDesign.

Documentation relative aux scripts d'InCopy

Vous pouvez télécharger la documentation relative aux scripts d'InCopy à partir de la page de la documentation pour développeur d'InDesign (mentionnée ci-dessus). Cette documentation se présente comme suit :

- ◆ *Guide des scripts d'Adobe InCopy* (versions AppleScript, JavaScript et VBScript) : ce document propose un didacticiel élémentaire suivi d'un exposé de sujets plus complexes concernant les scripts InCopy. Tous les scripts d'apprentissage présentés sont regroupés dans une archive ZIP, et donc accessibles sans un copier-coller à partir du fichier PDF. La plupart des scripts figurant dans le texte sont des extraits illustrant une propriété, méthode ou technique particulière.
- ◆ *JavaScript Tools and Features (Outils et fonctions JavaScript)* : ce document explique comment utiliser l'utilitaire ExtendScript Toolkit pour développer des scripts JavaScript, comment concevoir des interfaces utilisateur avec le module ScriptUI, comment utiliser les objets de type fichier et dossier, sans oublier d'autres fonctionnalités propres au langage ExtendScript (l'équivalent du langage JavaScript chez Adobe).

Il n'existe pas de fichier PDF *Guide de référence pour les scripts* pour cette version. A défaut, utilisez l'afficheur de modèle d'objet fourni avec votre application d'édition de scripts (comme indiqué dans le *Guide des scripts d'Adobe InCopy*).

Les scripts d'exemple InCopy sont installés par défaut. Ils apparaissent dans le panneau Scripts (commande Fenêtre > Utilitaires > Scripts).

Installation des scripts de la documentation relative aux scripts

Indépendamment des scripts d'exemple, tous les extraits de scripts présentés dans la documentation sont téléchargeables à partir de la page de la documentation pour développeur d'InDesign.

Après avoir téléchargé l'archive ZIP et avoir extrait les scripts de l'archive, déplacez le ou les dossiers correspondant au langage que vous voulez utiliser (AppleScript, JavaScript ou VBScript) dans le dossier que vous aurez créé dans votre panneau Scripts. Pour en savoir plus sur l'installation des scripts, voir le *Guide des scripts d'Adobe InCopy*.

Exécution de scripts

Pour exécuter un script, cliquez deux fois sur son nom dans le panneau Scripts.

Vous pouvez exécuter des scripts VBScripts (dotés de l'extension de fichier .vbs) ou JavaScripts (dotés de l'extension de fichier .jsx) sous Windows, et des scripts AppleScripts (portant l'extension de fichier .applescript) ou JavaScripts sous Mac OS.

Scripts d'exemple

Prenez soin d'enregistrer votre travail avant d'exécuter un script d'exemple pour la première fois.

Avant d'utiliser des scripts d'exemple avec des documents InCopy importants, testez-les afin de bien comprendre leur fonction.

InCopy inclut les scripts d'exemple suivants.

Nom du script	Description	Pour plus de détails, voir page...
CreateCharacterStyle	Définit un style de caractère complet d'après le texte sélectionné.	2
ExportAllStories	Exporte tous les articles d'un document dans une série de fichier texte.	2
FindChangeByList	Réalise une série d'opérations courantes de recherche/remplacement de texte par une simple lecture d'un fichier texte séparé par des tabulations.	3
SortParagraphs	Trie par ordre alphabétique les paragraphes d'une sélection.	3
TabUtilities	Applique des taquets de tabulation et des retraits au texte sélectionné.	3

Descriptions de scripts

Cette section décrit succinctement chacun des scripts d'exemple.

CreateCharacterStyle

Définit un style de caractère complet d'après le texte sélectionné.

Thèmes illustrés :

- ◆ Traitement des objets d'une sélection
- ◆ Création d'un style de caractère
- ◆ Récupération des attributs de mise en forme de texte à partir d'un objet texte
- ◆ Remplissage des propriétés de style de caractère

ExportAllStories

Exporte tous les articles d'un document dans une série de fichier texte.

Thèmes illustrés :

- ◆ Exportation de texte
- ◆ Méthodes et objets de fichier/dossier JavaScript (JavaScript uniquement)
- ◆ Création de noms de fichier d'après les attributs d'identification
- ◆ Création d'une interface utilisateur

FindChangeByList

Réalise une série d'opérations courantes de recherche/remplacement de texte par une simple lecture d'un fichier texte séparé par des tabulations.

Pour plus de détails, examiner le début du script ou du fichier `FindChangeList.txt`. Vous pouvez ajouter vos opérations de recherche/remplacement au fichier `FindChangeList.txt` ou créer votre propre fichier.

Thèmes illustrés :

- ◆ Utilisation des fichiers et des dossiers
- ◆ Recherche d'un fichier/dossier lié au script actif
- ◆ Utilisation des méthodes de recherche/remplacement de texte (texte, GREP et glyphe)
- ◆ Lecture des données d'un texte séparé par des tabulations à partir d'un fichier texte

SortParagraphs

Trie par ordre alphabétique les paragraphes d'une sélection.

Thèmes illustrés :

- ◆ Méthode de déplacement des objets texte
- ◆ Tri par permutation simple

TabUtilities

Applique des taquets de tabulation et des retraits au texte sélectionné.

Thèmes illustrés :

- ◆ Utilisation des taquets de tabulation et des retraits
- ◆ Récupération des positions d'une page à partir des objets texte
- ◆ Récupération de la colonne de texte dans laquelle est positionné le curseur
- ◆ Création d'une interface utilisateur

Problèmes connus liés à l'élaboration de scripts InCopy

Emplacement des scripts de lancement JavaScript

Les scripts de lancement utilisateur doivent être placés au même emplacement que les `scripts de lancement InCopy` (où ils s'exécutent une seule fois à chaque lancement de l'application), et non à l'emplacement où se trouvent les scripts d'initialisation de moteur `ExtendScript` (où ils sont exécutés à chaque initialisation d'un moteur).

Pour exécuter les scripts au lancement du logiciel InDesign, placez-les dans le sous-dossier `Startup Scripts` (Scripts de lancement) du dossier `Scripts`, situé dans le dossier d'application InCopy. (Si ce dossier n'existe pas, créez-le.)

Les scripts exécutés en dehors d'InCopy ne peuvent pas créer de moteurs ExtendScript permanents (JavaScript uniquement)

Comme indiqué au chapitre 2, « Scripts », du *Guide des scripts d'Adobe InCopy : JavaScript*, les scripts ExtendScript peuvent créer des instances permanentes du moteur ExtendScript. Les fonctions et variables définies dans le moteur permanent peuvent être utilisées par d'autres scripts qui exécutent ce moteur. Toutefois, pour créer un moteur ExtendScript permanent, le script doit être exécuté à partir du panneau Scripts d'InCopy. En effet, l'exécution du script à partir de l'utilitaire Outils ExtendScript ou au moyen de BridgeTalk à partir d'une autre application ne crée pas le moteur permanent.

Les détecteurs (ou guetteurs) d'événements ajoutés ou supprimés pendant la propagation d'événements ne sont pas gérés conformément à la spécification du consortium W3C

La *spécification du modèle objet de document (DOM) niveau 2 Events* (voir le lien <http://www.w3.org/TR/DOM-Level-2-Events/Overview.html>) stipule ce qui suit :

« Si on ajoute un guetteur `EventListener` à une cible `EventTarget` alors qu'elle traite un événement, ce guetteur ne sera pas déclenché par les actions courantes mais pourra l'être lors d'une étape suivante du flux d'événement, telle que la phase de bouillonnement. »

Ainsi que :

« Si on retire un guetteur `EventListener` d'une cible `EventTarget` alors qu'elle traite un événement, ce guetteur ne sera pas déclenché par les actions courantes. On ne peut plus invoquer les guetteurs `EventListener` après leur retrait. »

En programmation de scripts pour InCopy, les détecteurs d'événements ajoutés à une cible d'événement au cours de la propagation d'un événement ne sont pas déclenchés pendant toute la durée de l'événement. Les détecteurs d'événements supprimés d'une cible d'événement pendant la propagation d'un événement continuent d'être déclenchés par l'événement (par exemple, ils sont retirés une fois le traitement de l'événement terminé).

L'utilitaire Outils ExtendScript n'affiche pas la liste des scripts InCopy (Mac OS uniquement)

Par défaut, pendant son lancement, l'utilitaire Outils ExtendScript ne spécifie pas d'application cible. En conséquence, la liste des scripts disponibles dans le panneau Scripts (de l'utilitaire Outils ExtendScript et non du logiciel InCopy) n'est pas peuplée avec les scripts InCopy disponibles. Définissez InCopy comme application cible pour que l'utilitaire Outils ExtendScript charge les scripts dans le panneau Scripts.