

# ADOBE® AFTER EFFECTS® CS3 PROFESSIONAL

## GUIDE DES SCRIPTS



© Copyright 2007 Adobe Systems Incorporated. Tous droits réservés.

### Guide des scripts d'After Effects® d'Adobe® Creative Suite 3

Remarque : toutes les informations contenues dans le présent document sont la propriété d'Adobe Systems Incorporated. Toute reproduction ou diffusion du présent document (que ce soit au format électronique ou sous une forme imprimée), en tout ou partie, sous quelque forme ou par quelque moyen que ce soit, électronique, mécanique, photocopie, enregistrement ou autre, sans l'autorisation écrite préalable d'Adobe Systems Incorporated, est strictement interdite. Le logiciel décrit dans ce document est cédé sous licence et ne peut être utilisé et/ou copié que conformément aux conditions de cette licence.

Ce document et les informations qu'il contient sont fournis TELS QUELS, peuvent être modifiés sans préavis et ne constituent pas un engagement de la part d'Adobe Systems Incorporated. Adobe Systems Incorporated ne peut être tenu pour responsable des erreurs ou des inexactitudes, n'apporte aucune garantie (expresse, implicite ou statutaire) concernant cette publication et rejette expressément toute garantie de qualité marchande, d'adaptation à un objectif donné et de non-respect des droits de tiers.

Toute référence à des noms de société dans les exemples n'a qu'une valeur explicative et n'a pas pour objet de citer des organisations réelles.

Adobe, le logo Adobe, After Effects, Photoshop et Bridge sont des marques ou des marques déposées d'Adobe Systems Incorporated aux Etats-Unis et/ou dans d'autres pays.

Apple, Mac, Macintosh et Mac OS sont des marques d'Apple Computer, Inc., déposées aux Etats-Unis et dans d'autres pays. Microsoft et Windows sont des marques ou des marques déposées de Microsoft Corporation aux Etats-Unis et dans d'autres pays. JavaScript et toutes les marques contenant Java sont des marques ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. UNIX est une marque déposée de The Open Group.

Toutes les autres marques appartiennent à leurs propriétaires respectifs.

Si ce guide accompagne un logiciel assorti d'un contrat de licence, le guide et le logiciel dont il traite sont cédés sous licence et ne peuvent être copiés ou utilisés que conformément aux conditions générales de cette licence. Sauf autorisation spécifiée dans la licence, aucune partie de ce guide ne peut être reproduite, enregistrée ou transmise sous quelque forme que ce soit, par quelque moyen que ce soit, électronique, mécanique, enregistrement ou autre, sans l'autorisation écrite préalable d'Adobe Systems Incorporated. Nous attirons votre attention sur le fait que le contenu du présent guide est protégé par les lois sur la propriété intellectuelle et ce, même s'il n'accompagne pas un logiciel assorti d'un contrat de licence.

Les informations contenues dans ce guide sont données à titre purement indicatif. Elles peuvent être modifiées sans préavis et ne constituent pas un engagement de la part d'Adobe Systems Incorporated. Adobe Systems Incorporated ne peut être tenu pour responsable des erreurs ou des inexactitudes apparaissant dans les informations fournies par ce guide.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, Etats-Unis.

# Présentation

Le *Guide des scripts d'After Effects* explique comment gérer vos projets After Effects en créant des procédures à l'aide de scripts. Cet ensemble de fonctions est disponible dans Adobe® After Effects® CS3 Professional.

La création de script au niveau système permet de simplifier votre structure de rendu et d'éviter les nombreuses opérations de pointer-cliquer répétitives. Si vous avez déjà eu l'occasion d'utiliser des expressions ou d'autres techniques de type JavaScript pour l'animation, ou de travailler avec la création de script système dans AppleScript ou Visual Basic, vous percevrez rapidement l'étendue des possibilités offertes par la création de script dans l'application After Effects. Avec un peu de pratique et une bonne connaissance du langage JavaScript, vous pouvez prendre le contrôle de votre structure graphique.

## Si vous êtes novice en matière de création de script

After Effects est un outil visuel comportant une interface graphique utilisateur dont les éléments vous sont familiers, notamment les menus, les panneaux et les icônes. Cette interface est le moyen le plus accessible de travailler avec l'application. La création de script est conçue pour les situations dans lesquelles cette méthodologie implique une répétition fastidieuse ou de lourdes opérations de recherche et de tri qui pourraient être automatisées. La création de script permet d'automatiser les tâches rébarbatives et répétitives. Elle est également utile pour tirer parti de la puissance du rendu en réseau dans les cas où le dossier de contrôle est moins puissant (et moins pratique à configurer). Pour obtenir des exemples d'actions pouvant être effectuées par les scripts, reportez-vous à la section « Exemples », page 175.

Si vous êtes novice en matière de création de script, consultez le manuel *Adobe Introduction to Scripting*, qui présente les concepts de base de la création de script et décrit les différents langages disponibles, y compris JavaScript. JavaScript et les autres langages de création de script sont orientés objet ; ce manuel expose ainsi les concepts de base de la programmation orientée objet et des modèles d'objet de document.

Même si vous n'avez aucunement l'intention d'apprendre le langage JavaScript, vous pouvez néanmoins exploiter la puissance de la création de script par le biais de solutions tierces telles que Rush Network Render Queue, une interface graphique utilisateur permettant de configurer des rendus distribués depuis tout ordinateur du réseau sans devoir les configurer sur les machines de manière individuelle.

Les utilisateurs qui créent des scripts peuvent par ailleurs partager leurs scripts avec les autres utilisateurs pour une mise en commun efficace des ressources. Il est fréquent que les grands studios emploient ce type d'utilisateurs maîtrisant la création de script. Les utilisateurs plus novices peuvent quant à eux visiter des forums tels que ceux accessibles à l'adresse suivante : [www.adobeforums.com](http://www.adobeforums.com).

## A propos de ce guide

Ce guide est destiné aux utilisateurs chargés de la gestion d'une structure graphique (laquelle peut inclure d'autres applications scriptables) et qui souhaitent créer des scripts pour personnaliser After Effects.

Cette fonctionnalité est également offerte par des solutions tierces de gestion de rendu en réseau. Ces produits comportent des éléments logiciels conçus pour faciliter la gestion de ce processus. Il est ainsi possible de tirer profit de cette fonctionnalité sans devoir procéder à une édition manuelle des scripts.

Le cœur d'une application scriptable est le modèle d'objet. Lorsque vous utilisez Adobe After Effects, vous créez des projets, des compositions et des éléments de file d'attente de rendu, ainsi que tous les éléments qu'ils contiennent, c'est-à-dire le métrage, les images, les solides, les calques, les masques, les effets et les propriétés. En terminologie de création de script, chacun de ces éléments est un objet. Ce guide décrit les objets JavaScript définis pour des projets After Effects.

Une grande partie de ce que la création de script permet de réaliser peut également être effectuée par le biais de l'interface utilisateur After Effects. Une connaissance approfondie de l'application elle-même est, par conséquent, essentielle pour savoir comment utiliser cette fonctionnalité.

Le modèle d'objet d'After Effects est composé d'un projet, d'éléments, de compositions, de calques et d'éléments de file d'attente de rendu. Chaque objet possède ses propres attributs particuliers et chaque objet d'un projet After Effects possède sa propre identité (bien que tous ne soient pas accessibles pour l'écriture de scripts). Vous devez maîtriser le modèle d'objet d'After Effects pour pouvoir créer des scripts.

La création de script dans After Effects repose sur ECMAScript (plus précisément, sur la 3e édition de la norme ECMA-262). Pour plus d'informations sur cette norme, consultez le site [www.ecma-international.org](http://www.ecma-international.org). Pour exploiter le potentiel de la création de script, vous devez également comprendre la rédaction de script au niveau système (pour une intégration avec AppleScript ou l'application en ligne de commande Terminal sur Mac OS et les scripts de ligne de commande sur les systèmes Windows) et posséder une connaissance préalable de JavaScript.

*REMARQUE : les objets JavaScript habituellement appelés « propriétés » sont appelés « attributs » dans ce guide, afin d'éviter toute confusion avec la définition du terme « propriété » propre à After Effects (une valeur animable d'un effet, d'un masque ou d'une transformation dans un calque spécifique).*

## Expressions

Bien que les expressions et l'interface de création de script d'After Effects utilisent toutes les deux le langage JavaScript et aient accès à chacune des propriétés des calques, il s'agit de deux entités complètement différentes. Alors que les expressions n'ont pas accès aux informations des scripts (comme les variables et les fonctions), un script peut être rédigé pour créer ou éditer une expression.

Etant donné que les expressions et la création de script utilisent toutes les deux le langage JavaScript, la connaissance de l'une de ces fonctions facilite la compréhension de l'autre.

## Equations

Les équations ne font plus partie d'After Effects, cette fonctionnalité ayant été remplacée par la création de script et les expressions. Tous les opérateurs mathématiques et logiques communs à ECMAScript sont disponibles pour la création de script.

Par exemple, avec les expressions, il est possible de simuler le rebond d'une balle en appliquant des règles mathématiques à un calque « balle ». Avec la création de script, vous pouvez créer une interface utilisateur complète permettant d'animer une balle rebondissante et un calque d'ombre au moyen de critères entrés par l'utilisateur.

## Edition de scripts

After Effects inclut un éditeur JavaScript. Pour le démarrer, choisissez Fichier > Scripts > Ouvrir l'éditeur de script. Cet éditeur et débogueur de script, appelé ExtendScript Toolkit, fournit une interface conviviale vous permettant de créer et de tester vos propres scripts.

Vous pouvez utiliser tout éditeur de texte pour créer, éditer et enregistrer des scripts, mais il est recommandé de choisir une application qui n'ajoute pas automatiquement des informations d'en-tête lors de l'enregistrement des fichiers et qui enregistre avec l'encodage Unicode (UTF-8).

- Les applications Windows utiles pour l'édition de scripts incluent EM Editor ou le Bloc-notes (assurez-vous de définir le codage UTF-8 dans les options d'enregistrement).
- Les applications Mac OS utiles pour l'édition de scripts incluent BBEdit ou TextEdit intégré à OS X (assurez-vous de définir Unicode [UTF-8] comme type d'enregistrement dans les préférences).

### Format JSX d'ExtendScript

After Effects prend en charge ExtendScript, l'implémentation étendue de JavaScript d'Adobe. ExtendScript est utilisé par toutes les applications Adobe qui disposent d'une interface de script. Outre l'implémentation du langage JavaScript conformément aux spécifications ECMA 262 et E4X ECMA 357, ExtendScript offre des fonctionnalités et utilitaires supplémentaires :

**ExtendScript Toolkit :** Pour vous aider à développer, déboguer et tester vos scripts, ExtendScript offre un environnement interactif de développement et de test, ExtendScript Toolkit. Cet environnement définit également un objet de débogage global, l'objet dollar (\$), ainsi qu'un utilitaire de génération de rapports pour les éléments ExtendScript, l'interface Reflection d'ExtendScript.

**Objets File et Folder :** Dans la mesure où la syntaxe des chemins varie considérablement d'un système d'exploitation à l'autre, Adobe ExtendScript définit des objets File et Folder permettant d'offrir un accès indépendant de la plate-forme au système de fichiers sous-jacent.

**Module d'interface utilisateur ScriptUI :** Le module ScriptUI d'ExtendScript permet de créer des éléments d'interface utilisateur et d'interagir avec ces derniers. ScriptUI offre un modèle d'objet pour les fenêtres et les éléments de contrôle d'interface utilisateur que vous pouvez utiliser pour créer une interface utilisateur pour vos scripts.

**Outils et utilitaires :** ExtendScript comporte également des outils et des fonctionnalités tels qu'un utilitaire de localisation qui permet de spécifier des chaînes d'interface utilisateur en plusieurs langues, ainsi que des fonctions globales d'affichage de messages brefs dans des boîtes de dialogue (`alert`, `confirm` et `prompt`).

**Communication interapplication :** ExtendScript met en place un environnement commun de création de script pour l'ensemble des applications Adobe et permet les communications interapplication par le biais des scripts.

**Communication externe :** ExtendScript offre un objet `Socket` qui vous permet de communiquer avec des systèmes distants à partir de vos scripts After Effects.

Ces fonctionnalités, ainsi que plusieurs autres, sont décrites en détail dans le manuel *JavaScript Tools Guide*, fourni avec After Effects et disponible à partir du Pôle de développement Adobe (<http://www.adobe.fr/devnet/>).

Les fichiers de script ExtendScript portent l'extension `.jsx`, une variation de l'extension `.js` standard qui caractérise normalement les fichiers JavaScript. Les scripts After Effects doivent être dotés de l'extension de fichier `.jsx` pour pouvoir être correctement reconnus par l'application. Tout fichier texte au format UTF-8 et portant l'extension `.jsx` est admis en tant que fichier ExtendScript.

Vous pouvez utiliser ExtendScript Toolkit pour exporter une version binaire d'un fichier ExtendScript portant l'extension `.jsxbin`. Ce fichier binaire ne sera pas forcément utilisable avec toutes les fonctionnalités d'intégration de script proposées par After Effects.

## Activation des fonctionnalités complètes de création de script

Pour des raisons de sécurité, les fonctionnalités de création de script opérant en dehors de l'application After Effects (telles que l'ajout et la suppression de fichiers et de dossiers sur des volumes, ou l'accès au réseau) sont désactivées par défaut.

Pour activer ces fonctionnalités, choisissez Préférences > Général, puis sélectionnez « Autoriser les scripts à écrire des fichiers et à accéder au réseau ». Vous pouvez alors effectuer les opérations suivantes :

- écrire dans des fichiers ;
- créer des dossiers et définir le dossier actif ;
- créer une connexion par socket (pour plus d'informations sur cet utilitaire JavaScript, consultez le manuel *JavaScript Tools Guide*).

Adobe met à votre disposition un débogueur JavaScript complet, appelé ExtendScript Toolkit. Ce dernier est désactivé par défaut de façon à ce que les utilisateurs non expérimentés ne le rencontrent pas. Lors de l'édition ou de l'écriture de scripts, Toolkit permet de diagnostiquer les problèmes de script plus rapidement. Pour activer Toolkit sur la machine locale en cas d'erreur de script, choisissez Préférences > Général, puis cochez la case Activer le débogage JavaScript. Pour obtenir des informations détaillées sur ExtendScript Toolkit, consultez le manuel *JavaScript Tools Guide*.

Toolkit ne fonctionne que lors de l'exécution d'un script, pas avec les expressions, même si les expressions utilisent également JavaScript.

## Accès aux scripts et écriture des scripts

Pour créer et modifier des scripts pour After Effects, vous pouvez utiliser ExtendScript Toolkit ou une application externe d'édition de texte créant des fichiers avec l'encodage de texte Unicode UTF-8. Evitez les applications telles que Microsoft Word qui ajoutent par défaut des informations d'en-tête aux fichiers ; ces informations créent des erreurs de ligne 0 dans les scripts, ce qui empêche leur exécution.

Un script peut résider à tout endroit, mais pour apparaître dans le menu Scripts, il doit être sauvegardé dans le dossier Scripts, lui-même situé dans le dossier de l'application After Effects.

L'application n'intègre pas de méthode permettant l'enregistrement dans un script d'une série d'actions réalisées dans After Effects, comme cela est possible avec les scripts Adobe Photoshop®. Les scripts sont créés en dehors d'After Effects, puis exécutés à l'intérieur ou à l'extérieur de ce programme au moyen d'une ligne de commande, d'ExtendScript Toolkit ou d'un logiciel de gestion de rendu tiers.

### Menu et dossier Scripts

Les scripts After Effects résident dans le dossier Scripts, situé dans le même dossier que votre fichier d'application After Effects. Bien qu'un fichier de script puisse résider à un autre endroit, seuls les scripts contenus dans ce dossier Scripts lors du démarrage de l'application figurent automatiquement dans le menu Scripts.

Pour exécuter un script n'apparaissant pas dans le menu Scripts, sélectionnez Fichier > Scripts > Exécuter le fichier de script, puis choisissez le script dans la boîte de dialogue Ouvrir. Une autre solution consiste à envoyer un script à After Effects à partir d'ExtendScript Toolkit, d'une ligne de commande (sous Windows) ou d'AppleScript (sous Mac OS).

Pour apparaître dans la boîte de dialogue Ouvrir, votre script doit être doté de l'extension de fichier `.jsx`.

### Dossiers Shutdown (arrêt) et Startup (démarrage)

Le dossier Scripts contient deux sous-dossiers appelés Startup et Shutdown. After Effects exécute automatiquement, dans l'ordre alphabétique, les scripts contenus dans ces dossiers lors du démarrage et de la fermeture de l'application.

Vous pouvez placer dans le dossier Startup les scripts que vous souhaitez exécuter au démarrage de l'application. Une fois l'application initialisée, ces scripts seront exécutés et tous les modules seront chargés.

La création de script partageant un environnement global, tout script exécuté au démarrage peut définir des variables et des fonctions disponibles pour tous les scripts. Dans tous les cas de figure, les variables et les fonctions, une fois définies en exécutant un script les contenant, sont conservées dans les scripts suivants au cours de la même session After Effects. Une fois l'application fermée, toutes ces variables et fonctions définies au niveau global sont supprimées. Il est essentiel de nommer les variables de façon unique afin de ne pas remplacer par erreur une variable globale, censée être préservée pendant toute une session.

Des attributs peuvent également être ajoutés aux objets existants tels que l'objet Application (voir la section « Objet Application », page 19) pour étendre l'application à d'autres scripts.

Les scripts contenus dans le dossier Shutdown sont exécutés au moment de la fermeture de l'application. Ces exécutions sont réalisées une fois le projet fermé, mais avant la fermeture de toute autre application.

### Menu Fenêtre et dossier ScriptUI Panels (panneaux ScriptUI)

Dans le dossier Scripts, vous pouvez créer un autre dossier nommé ScriptUI Panels. Utilisez ce dossier pour les scripts dont l'interface utilisateur apparaît dans un panneau natif (et non dans une palette flottante, une boîte de dialogue ou une fenêtre). L'avantage d'un panneau réside dans sa capacité à être ancré dans d'autres panneaux, comme les panneaux Projet, Composition et Navigation, ce qui lui permet de paraître ainsi mieux intégré à l'application. Tout comme les panneaux natifs, les scripts du dossier ScriptUI Panels sont accessibles à partir du menu Fenêtre.

Au lieu de créer un objet Window et d'y intégrer des commandes, les scripts du dossier ScriptUI Panels utilisent l'objet « this » représentant le panneau. Par exemple, le code suivant ajoute un bouton à un panneau :

```
var myPanel = this;
myPanel.add("button", [10, 10, 100, 30], "Tool #1");
myPanel.show();
```

Si votre script crée son interface utilisateur dans une fonction, vous ne pouvez pas utiliser l'objet « this » puisqu'il se rapporte à la fonction elle-même et non au panneau. Dans ce cas, vous devez utiliser l'objet « this » en tant qu'argument de votre fonction. Par exemple :

```
function createUI(thisObj) {
    var myPanel = thisObj;
    myPanel.add("button", [10, 10, 100, 30], "Tool #1");
    return myPanel;
}
var myToolsPanel = createUI(this);
myToolsPanel.show();
```

Vous ne pouvez pas utiliser la commande de menu Fichier > Scripts > Exécuter le fichier de script pour exécuter un script renvoyant à un objet « this ». Pour permettre à votre script de fonctionner avec un objet Window (accessible à partir du menu Fichier > Scripts) ou avec un panneau natif (accessible à partir du menu Fenêtre), vérifiez que l'objet « this » est bien un objet Panel. Par exemple :

```
function createUI(thisObj) {
    var myPanel = (thisObj instanceof Panel) ? thisObj : new Window("palette", "My Tools",
        [100, 100, 300, 300]);
    myPanel.add("button", [10, 10, 100, 30], "Tool #1");
    return myPanel;
}
var myToolsPanel = createUI(this);
myToolsPanel.show();
```

## Envoi d'un script à After Effects à partir du système

Si vous savez comment exécuter un script à partir de la ligne de commande sous Windows ou avec AppleScript, vous pouvez envoyer un script directement à l'application After Effects ouverte pour que celle-ci l'exécute automatiquement.

### Inclusion de la création de script After Effects dans une ligne de commande (Windows)

Voici plusieurs exemples de commandes pour la ligne de commande Windows qui envoient un script After Effects à l'application sans passer par l'interface utilisateur After Effects.

Dans le premier exemple, vous devez copier et coller votre script After Effects directement dans la ligne de commande puis l'exécuter. Le texte du script s'affiche entre guillemets après la commande `afterfx.exe -s` :

```
afterfx.exe -s "alert("You just sent an alert to After Effects")"
```

Une autre solution consiste à spécifier l'emplacement du fichier JSX à exécuter. Par exemple :

```
afterfx.exe -r c:\myDocuments\Scripts\yourAEScriptHere.jsx  
afterfx.exe -r "c:\myDocuments\Scripts\Script Name with Spaces.jsx"
```

### Inclusion de la création de script After Effects dans un AppleScript (Mac OS)

Voici trois exemples de scripts AppleScript envoyant un fichier JSX existant contenant un script After Effects à l'application sans utiliser l'interface utilisateur After Effects pour exécuter le script.

Dans le premier exemple, vous devez copier votre script After Effects directement dans l'éditeur de script puis l'exécuter. Le texte du script s'affiche entre guillemets après la commande `DoScript` ; par conséquent, tout guillemet contenu dans cette chaîne doit être précédé d'une barre oblique inversée servant de caractère d'échappement, tel qu'indiqué ci-dessous :

```
tell application "Adobe After Effects CS3"  
  DoScript "alert(\\"You just sent an alert to After Effects\\")"  
end tell
```

Une autre solution consiste à afficher une boîte de dialogue demandant l'emplacement du fichier JSX à exécuter, tel qu'indiqué ci-dessous :

```
set theFile to choose file  
tell application "Adobe After Effects CS3"  
  DoScript theFile  
end tell
```

Enfin, ce script est peut-être le plus utile lorsque vous travaillez directement sur l'édition d'un script JSX et souhaitez l'envoyer à After Effects pour le tester ou l'exécuter. Pour l'utiliser efficacement, vous devez spécifier l'application contenant le fichier JSX ouvert (dans cet exemple, il s'agit de TextEdit) ; si vous ne connaissez pas le nom exact de l'application, tapez ce qui s'en approche le plus à la place de « TextEdit », et AppleScript vous invitera à la localiser.

Mettez simplement en évidence le texte de script que vous souhaitez exécuter, puis activez cet AppleScript :

```
(*  
This script sends the current selection to After Effects as a script.  
*)
```

```
tell application "TextEdit"  
    set the_script to selection as text  
end tell
```

```
tell application "Adobe After Effects CS3"  
    activate  
    DoScript the_script  
end tell
```

Pour plus d'informations sur l'utilisation d'AppleScript, reportez-vous au manuel *AppleScript: the Definitive Guide* de Matt Neuberg (éditions O'Reilly & Associates) ou *AppleScript 1-2-3* de Sal Soghoian (éditions Peachpit Press).

## Essais et dépannage

Tout script After Effects contenant une erreur empêchant son exécution complète génère un message d'erreur dans l'application. Ce message inclut des informations sur la nature de l'erreur et la ligne du script sur laquelle elle est survenue.

After Effects inclut par ailleurs un outil de débogage JavaScript. Pour plus de détails sur l'activation et l'utilisation du débogueur, consultez la section relative à ExtendScript Toolkit du manuel *JavaScript Tools Guide*.

## Autres ressources d'apprentissage de la création de script

Il existe de nombreuses ressources permettant d'approfondir ses connaissances de la création de script selon la norme ECMA.

Le moteur de création de script After Effects prend en charge la 3e édition de la norme ECMA-262, y compris ses conventions de notation et lexicales, ses types, objets, expressions et instructions.

Pour obtenir une liste complète des mots-clés et opérateurs inclus avec ECMAScript, reportez-vous au document ECMA-262.pdf, disponible à l'adresse suivante : [www.ecma-international.org/publications/standards/Ecma-262.htm](http://www.ecma-international.org/publications/standards/Ecma-262.htm).

Les ouvrages traitant de JavaScript 1.2 sont également utiles à la compréhension du fonctionnement de la création de script dans After Effects. L'ouvrage *JavaScript, The Definitive Guide* de David Flanagan (éditions O'Reilly) constitue, en quelque sorte, une bible pour les utilisateurs de JavaScript. Une autre source tout à fait recommandable est l'ouvrage *JavaScript: A Beginner's Guide* de John Pollock (éditions Osborne). Ces deux ouvrages contiennent des informations concernant en particulier les extensions de JavaScript pour les navigateurs Internet, tout en offrant également des explications détaillées des principes de base de la création de script.

Il existe également des ouvrages sur l'utilisation d'AppleScript et la création de scripts de ligne de commande Windows, ces deux solutions pouvant être utilisées pour envoyer des scripts à After Effects.

## Variables JavaScript

La création de script partageant un environnement global, tout script exécuté au démarrage peut définir des variables et des fonctions disponibles pour tous les scripts. Dans tous les cas de figure, les variables et les fonctions, une fois définies en exécutant un script les contenant, sont conservées dans les scripts suivants au cours de la même session After Effects. Une fois l'application fermée, toutes ces variables et fonctions définies au niveau global sont supprimées. Il est essentiel de nommer les variables de façon unique de façon à ne pas remplacer par erreur une variable globale, censée être préservée pendant toute une session.

## Mots-clés et syntaxe d'instructions JavaScript

Bien qu'il ne soit pas possible de fournir une ressource exhaustive décrivant l'utilisation de JavaScript, les tableaux ci-après présentent les mots-clés, les instructions, les opérateurs, la priorité et l'associativité.

Le tableau ci-dessous énumère et décrit tous les mots-clés et instructions reconnus par le moteur de création de script After Effects.

Tableau 1 Mots-clés et syntaxe d'instructions

Mot-clé/Instruction	Description
break	JavaScript standard ; quitte la boucle en cours d'exécution.
continue	JavaScript standard ; arrête l'exécution de l'itération de la boucle en cours.
case	Libellé utilisé dans une instruction <code>switch</code> .
default	Libellé utilisé dans une instruction <code>switch</code> lorsqu'un libellé <code>case</code> est introuvable.
do...while	Construction JavaScript standard. Similaire à la boucle <code>while</code> , excepté que l'évaluation de la condition de boucle est effectuée à la fin de la boucle.
false	Littéral correspondant au booléen <code>false</code> .
for	Construction de boucle JavaScript standard.
for...in	Construction JavaScript standard. Fournit un moyen aisé d'appliquer une boucle à travers les propriétés d'un objet.
function	Utilisé pour définir une fonction.
if/if...else	Constructions conditionnelles JavaScript standard.
new	Instruction de constructeur JavaScript standard.
null	Attribué à une variable, un élément de tableau ou une propriété d'objet pour indiquer qu'il ou elle ne contient pas de valeur autorisée.
return	Moyen JavaScript standard de renvoyer une valeur à partir d'une fonction ou de quitter une fonction.
switch	Moyen JavaScript standard d'évaluer une expression et de tenter de faire correspondre la valeur de l'expression à un libellé <code>case</code> .
this	Méthode JavaScript standard utilisée pour indiquer l'objet en cours.
true	Littéral correspondant au booléen <code>true</code> .
undefined	Indique que la variable, l'élément de tableau ou la propriété d'objet n'a pas encore reçu de valeur.
var	Syntaxe JavaScript standard utilisée pour déclarer une variable locale.
while	Construction JavaScript standard. Similaire à la boucle <code>do...while</code> , excepté que l'évaluation de la condition de boucle est effectuée au début de la boucle.
with	Construction JavaScript standard utilisée pour spécifier un objet à utiliser dans les instructions dont elle est suivie.

## Opérateurs JavaScript

Les tableaux ci-après énumèrent et décrivent tous les opérateurs reconnus par le moteur de création de script After Effects et indiquent la priorité et l'associativité de tous les opérateurs.

Tableau 2 Description des opérateurs

Opérateurs	Description
new	Alloue un objet.
delete	Annule l'allocation d'un objet.
typeof	Renvoie le type de données.
void	Renvoie la valeur indéfinie.
.	Membre de la structure.
[]	Élément de tableau.
()	Appel de fonction.
++	Pré- ou post-incrément.
--	Pré- ou post-décrément.
-	Négation ou soustraction unaire.
~	NOT au niveau bit.
!	NOT logique.
*	Produit.
/	Division.
%	Division modulo.
+	Addition.
<<	Décalage vers la gauche au niveau du bit.
>>	Décalage vers la droite au niveau du bit.
>>>	Décalage non signé vers la droite au niveau du bit.
<	Inférieur.
<=	Inférieur ou égal.
>	Supérieur.
>=	Supérieur ou égal.
==	Egal.
!=	Différent.
&	AND au niveau bit.
^	XOR au niveau bit.
	OR au niveau bit.
&&	AND logique.
	OR logique.
?:	Conditionnel (ternaire).

Opérateurs	Description
=	Attribution.
+=	Attribution avec l'opération d'addition.
-=	Attribution avec l'opération de soustraction.
*=	Attribution avec l'opération de multiplication.
/=	Attribution avec l'opération de division.
%=	Attribution avec l'opération de division modulo.
<<=	Attribution avec l'opération de décalage vers la gauche au niveau du bit.
>>=	Attribution avec opération de décalage vers la droite au niveau du bit.
>>>=	Attribution avec opération de décalage vers la droite non signée au niveau du bit.
&=	Attribution avec opération AND au niveau du bit.
^=	Attribution avec opération XOR au niveau du bit.
=	Attribution avec opération OR au niveau du bit.
,	Evaluation multiple.

Tableau 3 Priorité des opérateurs

Opérateurs (de la priorité la plus élevée à la plus faible)	Associativité
[], (), .	gauche à droite
new, delete, - (négation unaire), !, typeof, void, ++, --	droite à gauche
*, /, %	gauche à droite
+, - (soustraction)	gauche à droite
<<, >>, >>>	gauche à droite
<, <=, >, >=	gauche à droite
==, !=	gauche à droite
&	gauche à droite
^	gauche à droite
	gauche à droite
&&	gauche à droite
	gauche à droite
?:	droite à gauche
=, /=, %=, <<=, >>=, >>>=, &=, ^=,  =, +=, -=, *=	droite à gauche
,	gauche à droite

# Référence JavaScript

Ce chapitre énumère et décrit les classes, objets, méthodes, attributs et fonctions globales définis par After Effects.

Le moteur de création de script After Effects prend en charge ExtendScript, la version étendue de JavaScript d'Adobe, qui implémente la 3e édition de la norme ECMA-262, y compris ses conventions de notation et lexicales, ses types, objets, expressions et instructions. Pour obtenir une liste complète des mots-clés et opérateurs inclus avec ECMAScript, reportez-vous au document ECMA-262.pdf, disponible à l'adresse suivante : [www.ecma-international.org/publications/standards/Ecma-262.htm](http://www.ecma-international.org/publications/standards/Ecma-262.htm). Pour avoir une vue générale des mots-clés et instructions les plus courants disponibles en ECMA-262, reportez-vous à la section « Mots-clés et syntaxe d'instructions JavaScript », page 10.

## Modèle d'objet d'After Effects

Au fil de votre lecture de cette section de référence, qui reprend les objets par ordre alphabétique, vous pouvez vous reporter aux schémas suivants pour avoir une vue générale de la position des différents objets dans la hiérarchie et de leur correspondance dans l'interface utilisateur.

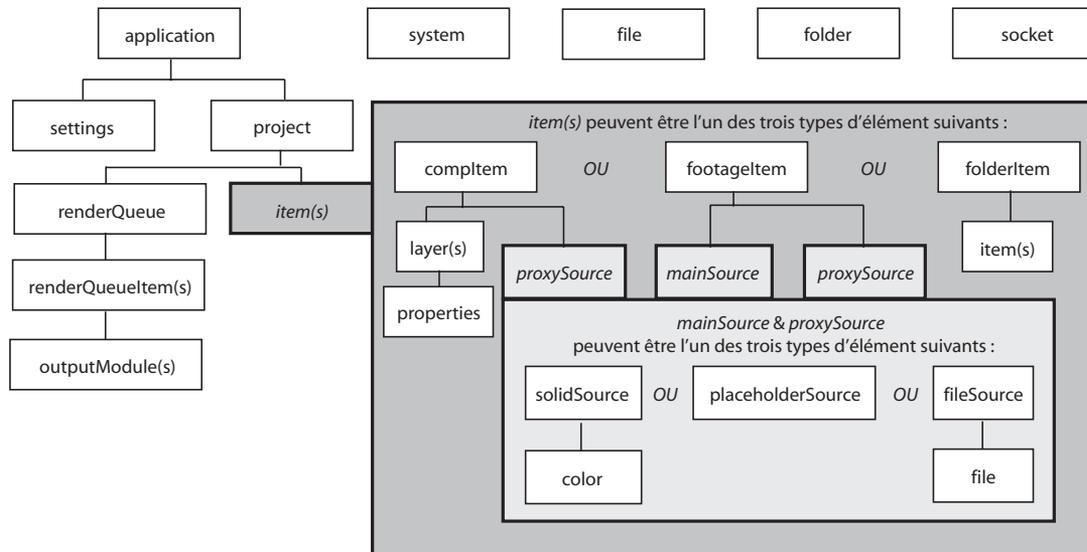
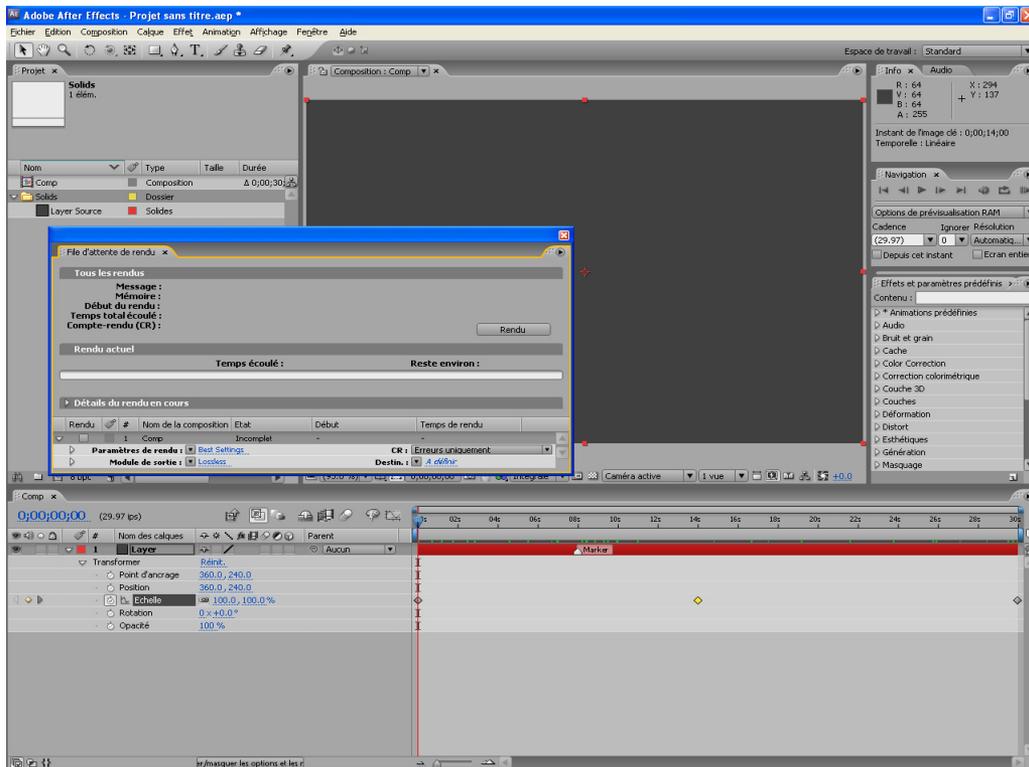


Schéma hiérarchique des principaux objets de création de scripts After Effects

Notez que les objets File, Folder et Socket sont définis par ExtendScript et présentés dans le manuel *JavaScript Tools Guide*. ExtendScript définit également le module ScriptUI, jeu d'objets de contrôle d'interface utilisateur et de fenêtres, auquel peuvent accéder les scripts After Effects. Ces objets de contrôle sont également présentés dans le manuel *JavaScript Tools Guide*.

La hiérarchie d'objets pour la création de script correspond à la hiérarchie de l'interface utilisateur.



L'application contient un panneau *Projet* qui affiche un *projet*. Le projet contient des *compositions*, contenant elles-mêmes des *calques*. La source d'un calque peut être un *fichier de métrage*, un *pseudo-élément* ou un *solide*, et figure également dans le panneau *Projet*. Chaque calque contient des paramètres appelés *propriétés*, qui peuvent contenir des *repères* et des *images clés*. La *file d'attente de rendu* contient des *éléments de file d'attente de rendu*, ainsi que des paramètres de rendu et des *modules de sortie*. Dans le contexte de la création de script, toutes ces entités sont représentées par des objets.

*REMARQUE* : dans ce guide, pour éviter toute ambiguïté, le terme « *attribut* » fait référence aux propriétés des objets JavaScript et le terme « *propriété* » ou « *propriété AE* » fait référence aux propriétés de calque After Effects.

## Liste des objets

Le tableau suivant répertorie tous les objets par ordre alphabétique et renvoie aux pages correspondantes.

Objet	Description
« Fonctions globales », page 16	Fonctions disponibles au niveau global. Elles vous permettent d'afficher du texte destiné au débogage de script et aident à convertir des secondes en images.
« Objet Application », page 19	Objet global unique, disponible via son nom ( <code>app</code> ). Il donne accès aux paramètres des objets et des applications au sein de l'application After Effects.
« Objet AVItem », page 32	Représente les fichiers audiovisuels importés dans After Effects.
« Objet AVLayer », page 39	Représente les calques qui contiennent des objets AVItem (calques de composition, de métrage, de solide, de texte et de son).
« Objet CameraLayer », page 50	Représente un calque de caméra dans une composition.
« Objet Collection », page 51	Associe un jeu d'objets ou de valeurs en tant que groupe logique, et donne accès à ces objets ou valeurs via un index.

Objet	Description
« Objet Compltem », page 52	Représente une composition, vous permet de la manipuler et d'obtenir des informations sur elle.
« Objet FileSource », page 60	Décrit le métrage issu d'un fichier.
« Objet FolderItem », page 62	Représente un dossier du panneau Projet.
« Objet FootageItem », page 64	Représente un élément de métrage importé dans un projet, qui apparaît dans le panneau Projet.
« Objet FootageSource », page 67	Décrit la source du fichier d'un métrage.
« Objet ImportOptions », page 73	Incorpore des options pour l'importation de fichiers dans After Effects.
« Objet Item », page 76	Représente un élément dans un projet qui apparaît dans le panneau Projet.
« Objet ItemCollection », page 79	Rassemble des éléments dans un projet.
« Objet KeyframeEase », page 81	Incorpore des valeurs de lissage d'images clés dans une propriété After Effects.
« Objet Layer », page 83	Classe de base pour les classes de calques.
« Objet LayerCollection », page 92	Rassemble des calques dans un projet.
« Objet LightLayer », page 97	Représente un calque de lumière dans une composition.
« Objet MarkerValue », page 98	Incorpore des valeurs de repère dans une propriété After Effects.
« Objet MaskPropertyGroup », page 102	Incorpore des attributs de masque dans un calque.
« Objet OMCollection », page 105	Rassemble des modules de sortie dans une file d'attente de rendu.
« Objet OutputModule », page 106	Représente un module de sortie pour une file d'attente de rendu.
« Objet PlaceholderSource », page 109	Décrit un pseudo-élément pour métrage.
« Objet Project », page 110	Représente un projet After Effects.
« Objet Property », page 119	Représente une propriété After Effects.
« Objet PropertyBase », page 141	Classe de base pour des classes de propriétés et de groupes de propriétés After Effects.
« Objet PropertyGroup », page 148	Représente un groupe de propriétés After Effects.
« Objet RenderQueue », page 153	Représente la file d'attente de rendu After Effects.
« Objet RenderQueueItem », page 156	Représente un élément de rendu dans une file d'attente de rendu.
« Objet RenderQueueItem », page 156	Rassemble des éléments de file d'attente de rendu dans une file d'attente de rendu.
« Objet RQItemCollection », page 163	Donne accès aux paramètres et préférences de l'application.
« Objet Shape », page 166	Incorpore la forme du contour d'un masque.
« Objet ShapeLayer », page 169	Représente un calque de forme dans une composition.
« Objet SolidSource », page 170	Décrit une couleur unie correspondant à la source d'un métrage.
« Objet System », page 171	Donne accès au système d'exploitation à partir de l'application.
« Objet TextDocument », page 173	Incorpore le texte dans un calque de texte.
« Objet TextLayer », page 174	Représente un calque de texte dans une composition.

## Fonctions globales

Ces fonctions, disponibles au niveau global, sont spécifiques à After Effects. Tout objet ou fonction JavaScript peut appeler ces fonctions. Elles vous permettent d'afficher du texte JavaScript dans une zone de petite taille (3 lignes) du panneau Info et de convertir des valeurs temporelles numériques en valeurs de type chaîne, et inversement.

Fonction globale	Description
<code>clearOutput()</code>	Efface le texte du panneau Info.
<code>currentFormatToTime()</code>	Convertit une valeur temporelle de type chaîne en valeur temporelle numérique.
<code>timeToCurrentFormat()</code>	Convertit une valeur temporelle numérique en valeur temporelle de type chaîne.
<code>write()</code>	Ecrit du texte dans le panneau Info, sans ajout de saut de ligne.
<code>writeLn()</code>	Ecrit du texte dans le panneau Info, en ajoutant un saut de ligne à la fin.

Des fonctions globales supplémentaires pour les E/S utilisateur standard (`alert`, `confirm` et `prompt`) et des fonctions statiques pour les E/S de fichier sont définies par `ExtendScript`. Pour plus d'informations, reportez-vous au *Guide de référence JavaScript* d'Adobe Bridge®.

*REMARQUE : les fonctions globales d'After Effects pour les boîtes de dialogue standard et les E/S de fichier sont toujours prises en charge dans cette version, mais sont désormais obsolètes et ne seront pas prises en charge dans les prochaines versions. Pour plus d'informations, reportez-vous à la documentation relative à After Effects 6.5.*

### Fonction globale `clearOutput()`

`clearOutput()`

#### Description

Efface la sortie dans le panneau Info.

#### Paramètres

Aucun.

#### Renvoi

Aucun.

### Fonction globale `currentFormatToTime()`

`currentFormatToTime(formattedTime, fps, isDuration)`

#### Description

Convertit une chaîne formatée d'instant d'image en nombre de secondes, selon une cadence d'image spécifiée. Par exemple, si l'instant de l'image formatée est 0:00:12 (le format de chaîne exact est déterminé par un paramètre de projet) et que la cadence d'image est de 24 i/s, l'instant sera 0,5 seconde (12/24). Si la cadence d'image est de 30 i/s, l'instant sera 0,4 seconde (12/30).

Si l'instant est une durée, le décompte des images commence à partir de zéro (0). Sinon, les images sont décomptées à partir de la première image du projet (voir la section « Attribut `Project displayStartFrame` », page 113).

**Paramètres**

<code>formattedTime</code>	Valeur d'instant de l'image, chaîne spécifiant un nombre d'images selon le format d'affichage de l'instant actuel du projet.
<code>fps</code>	Nombre d'images par seconde, valeur à virgule flottante.
<code>isDuration</code>	Facultatif. Lorsque ce paramètre est défini sur <code>true</code> , l'instant est une durée (mesurée à partir de l'image 0). Lorsqu'il est défini sur <code>false</code> (valeur par défaut), l'instant est calculé à partir de la première image du projet.

**Renvoi**

Valeur à virgule flottante, nombre de secondes.

**Fonction globale `timeToCurrentFormat()`**

`timeToCurrentFormat(time, fps, isDuration)`

**Description**

Convertit une valeur temporelle numérique (un nombre de secondes) en valeur temporelle d'image, c'est-à-dire en une chaîne formatée qui indique l'image correspondant à cet instant, selon la cadence spécifiée. Par exemple, si l'instant est 0,5 seconde et que la cadence d'image est de 24 i/s, l'image est 0:00:12 (lorsque le projet est défini sur `Display Timecode`). Si la cadence est de 30 i/s, l'image est 00:00:15. Le format de la chaîne de code temporel est défini par un paramètre de projet.

Si l'instant est une durée, le décompte des images commence à partir de zéro (0). Sinon, les images sont décomptées à partir de la première image du projet (voir la section « `Attribut Project displayStartFrame` », page 113).

**Paramètres**

<code>time</code>	Nombre de secondes, valeur à virgule flottante.
<code>fps</code>	Nombre d'images par seconde, valeur à virgule flottante.
<code>isDuration</code>	Facultatif. Lorsque ce paramètre est défini sur <code>true</code> , l'instant est une durée (mesurée à partir de l'image 0). Lorsqu'il est défini sur <code>false</code> (valeur par défaut), l'instant est calculé à partir de la première image du projet.

**Renvoi**

Chaîne dans le format d'affichage de l'instant actuel du projet.

**Fonction globale `write()`**

`write(text)`

**Description**

Ecrit la sortie dans le panneau Info, sans ajout de saut de ligne.

**Paramètres**

<code>text</code>	Chaîne à afficher. Elle est tronquée lorsqu'elle est trop longue pour être affichée dans le panneau Info.
-------------------	---

**Renvoi**

Aucun.

**Exemple**

```
write("This text appears in Info panel ");  
write("with more on same line.");
```

**Fonction globale writeLn()**

```
writeLn(text)
```

**Description**

Ecrit la sortie dans le panneau Info, en ajoutant un saut de ligne à la fin.

**Paramètres**

text	Chaîne à afficher.
------	--------------------

**Renvoi**

Aucun.

**Exemple**

```
writeln("This text appears on first line");  
writeln("This text appears on second line");
```

## Objet Application

app

### Description

Donne accès aux paramètres des objets et des applications au sein de l'application After Effects. L'objet global unique est toujours disponible via son nom, app.

Les attributs de l'objet Application fournissent l'accès aux objets spécifiques dans After Effects. Les méthodes de l'objet Application peuvent créer un projet, ouvrir un projet existant, contrôler le mode Dossier de contrôle, vider la mémoire et quitter l'application After Effects. Lors de la fermeture de l'application After Effects, celle-ci ferme le projet ouvert en demandant à l'utilisateur d'enregistrer ou d'ignorer les modifications, et crée un fichier de projet comme nécessaire.

### Attributs

Attribut	Référence	Description
project	« Attribut Application project », page 28 et « Objet Project », page 110	Projet After Effects en cours.
language	« Attribut Application language », page 24	Langue dans laquelle l'application est exécutée.
version	« Attribut Application version », page 30	Numéro de version de l'application After Effects.
buildName	« Attribut Application buildName », page 21	Nom de cette compilation de l'application.
buildNumber	« Attribut Application buildNumber », page 22	Numéro de cette compilation de l'application.
isWatchFolder	« Attribut Application isWatchFolder », page 24	Lorsque cet attribut est défini sur true, l'application locale est exécutée en mode Dossier de contrôle.
isRenderEngine	« Attribut Application isRenderEngine », page 24	Lorsque cet attribut est défini sur true, l'application After Effects locale est exécutée en tant que moteur de rendu.
settings	« Attribut Application settings », page 30 et « Objet RQItemCollection », page 163	Paramètres de l'application pouvant être définis par le biais de la création de script.
onError	« Attribut Application onError », page 26	Fonction de rappel appelée lorsqu'une erreur survient dans l'application.
exitCode	« Attribut Application exitCode », page 24	Code numérique d'état utilisé lors de l'exécution d'un script en externe (c'est-à-dire à partir d'une ligne de commande ou d'AppleScript). Zéro (0) indique qu'aucune erreur n'est survenue. Un nombre positif indique qu'une erreur est survenue lors de l'exécution du script.
exitAfterLaunchAndEval	« Attribut Application exitAfterLaunchAndEval », page 23	Lorsque cet attribut est défini sur true, l'application doit rester ouverte après l'exécution d'un script à partir de la ligne de commande sous Windows.
saveProjectOnCrash	« Attribut Application saveProjectOnCrash », page 29	Lorsque cet attribut est défini sur true, le projet est enregistré au cas où l'application viendrait à se fermer de façon inattendue.
memoryInUse	« Attribut Application memoryInUse », page 25	Mémoire utilisée par cette application.

**Méthodes**

Méthode	Référence	Description
newProject()	« Méthode Application newProject() », page 25	Crée un nouveau projet dans After Effects.
open()	« Méthode Application open() », page 26	Ouvre un projet ou une boîte de dialogue Ouvrir le projet.
quit()	« Méthode Application quit() », page 28	Ferme l'application.
watchFolder()	« Méthode Application watchFolder() », page 30	Démarre le mode Dossier de contrôle ; ne renvoie rien tant que ce mode n'est pas désactivé.
pauseWatchFolder()	« Méthode Application pauseWatchFolder() », page 27	Interrompt temporairement le processus de dossier de contrôle en cours.
endWatchFolder()	« Méthode Application endWatchFolder() », page 23	Met fin au processus de dossier de contrôle en cours.
purge()	« Méthode Application purge() », page 28	Purge un type ciblé d'informations mises en mémoire cache (reproduit les options de la commande Purger du menu Edition).
beginUndoGroup()	« Méthode Application beginUndoGroup() », page 21	Regroupe les actions qui la suivent en une seule étape pouvant être annulée.
endUndoGroup()	« Méthode Application endUndoGroup() », page 22	Termine un groupe d'annulation ; nécessaire seulement lorsqu'un script contient plusieurs groupes d'annulation.
beginSuppressDialogs()	« Méthode Application beginSuppressDialogs() », page 21	Commence la suppression des boîtes de dialogue dans l'interface utilisateur.
endSuppressDialogs()	« Méthode Application endSuppressDialogs() », page 22	Met fin à la suppression de boîtes de dialogue dans l'interface utilisateur.
setMemoryUsageLimits()	« Méthode Application setMemoryUsageLimits() », page 29	Définit les limites d'utilisation de mémoire, telles que spécifiées dans les préférences Mémoire et cache.
setSavePreferencesOnQuit()	« Méthode Application setSavePreferencesOnQuit() », page 30	Indique si les préférences doivent être enregistrées lors de la fermeture de l'application.
activate()	« Méthode Application activate() », page 20	Affiche la fenêtre principale d'After Effects au premier plan.
scheduleTask()	« Méthode Application scheduleTask() », page 29	Planifie un script JavaScript pour une exécution ultérieure.
cancelTask()	« Méthode Application cancelTask() », page 22	Annule une tâche planifiée.
parseSwatchFile()	« Méthode Application parseSwatchFile() », page 27	Charge un échantillon de couleur à partir d'un fichier Adobe Swatch Exchange (ASE).

**Méthode Application activate()**

app.activate()

**Description**

Ouvre la fenêtre principale de l'application, si elle apparaît sous forme réduite ou d'icône, et l'affiche au premier plan.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Méthode Application beginSuppressDialogs()**

```
app.beginSuppressDialogs()
```

**Description**

Lance la suppression des boîtes de dialogue d'erreur de script dans l'interface utilisateur. Pour reprendre l'affichage des boîtes de dialogue d'erreur, utilisez `endSuppressDialogs()`. Voir la section « Méthode Application `endSuppressDialogs()` », page 22.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Méthode Application beginUndoGroup()**

```
app.beginUndoGroup(undoString)
```

**Description**

Marque le début d'un groupe d'annulation qui permet à un script de regrouper de façon logique toutes ses actions sous la forme d'une seule action pouvant être annulée (via les options de menu Edition > Annuler/Rétablir). Pour marquer la fin du groupe, utilisez la méthode `endUndoGroup()`. (Voir la section « Méthode Application `endUndoGroup()` », page 22.)

Les paires `beginUndoGroup()` et `endUndoGroup()` peuvent être imbriquées. Les groupes dans les groupes deviennent partie intégrante du plus grand groupe et seront annulés correctement. Dans ce cas, les noms des groupes internes sont ignorés.

**Paramètres**

<code>undoString</code>	Texte qui apparaîtra pour la commande Annuler dans le menu Edition (c'est-à-dire « Annuler <undoString> »).
-------------------------	---

**Renvoi**

Aucun.

**Attribut Application buildName**

```
app.buildName
```

**Description**

Nom de la compilation d'After Effects en cours d'exécution. Adobe utilise ce nom en interne à des fins d'essais et de débogage.

**Type**

Chaîne ; lecture seule.

**Attribut Application buildNumber**`app.buildNumber`**Description**

Numéro de la compilation d'After Effects en cours d'exécution. Adobe utilise ce numéro en interne à des fins d'essais et de débogage.

**Type**

Entier ; lecture seule.

**Méthode Application cancelTask()**`app.cancelTask(taskID)`**Description**

Supprime la tâche spécifiée de la file d'attente des tâches planifiées pour une exécution ultérieure.

**Paramètres**

<code>taskID</code>	Entier qui identifie la tâche, telle qu'elle est renvoyée par la méthode <code>app.scheduleTask()</code> .
---------------------	--

**Renvoi**

Aucun.

**Méthode Application endSuppressDialogs()**`app.endSuppressDialogs(alert)`**Description**

Met fin à la suppression des boîtes de dialogue d'erreur de script dans l'interface utilisateur. Les boîtes de dialogue d'erreur s'affichent par défaut. Appelez cette méthode uniquement si la méthode `beginSuppressDialogs()` a été appelée préalablement. Voir la section « Méthode Application `beginSuppressDialogs()` », page 21.

**Paramètres**

<code>alert</code>	Booléen ; lorsque ce paramètre est défini sur <code>true</code> , les erreurs survenues après l'appel de la méthode <code>beginSuppressDialogs()</code> s'affichent dans une boîte de dialogue.
--------------------	---

**Renvoi**

Aucun.

**Méthode Application endUndoGroup()**`app.endUndoGroup()`**Description**

Marque la fin d'un groupe d'annulation commencé par la méthode `app.beginUndoGroup()`. Vous pouvez utiliser cette méthode pour placer une fin de groupe d'annulation au milieu d'un script si vous souhaitez utiliser plusieurs groupes d'annulation pour un seul script.

Si vous n'utilisez qu'un seul groupe d'annulation pour un script donné, vous n'avez pas besoin d'utiliser cette méthode. En son absence à la fin d'un script, le système fermera le groupe d'annulation automatiquement.

Le fait d'appeler cette méthode sans avoir défini de méthode `beginUndoGroup()` génère une erreur.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Méthode Application `endWatchFolder()`**

`app.endWatchFolder()`

**Description**

Met fin au mode Dossier de contrôle.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Voir aussi**

« Méthode Application `watchFolder()` », page 30

« Méthode Application `parseSwatchFile()` », page 27

« Attribut Application `isWatchFolder` », page 24

**Attribut Application `exitAfterLaunchAndEval`**

`app.exitAfterLaunchAndEval`

**Description**

Cet attribut n'est utilisé que lors de l'exécution d'un script à partir d'une ligne de commande sous Windows. Lorsque l'application est lancée à partir de la ligne de commande, si l'indicateur de ligne de commande `-r` ou `-s` est utilisé, l'application exécute un script (à partir d'un fichier ou d'une chaîne, respectivement).

Si cet attribut est défini sur `true`, After Effects quitte après l'exécution du script ; s'il est défini sur `false`, l'application reste ouverte.

Cet attribut n'a d'effet que si After Effects est exécuté à partir d'une ligne de commande Windows. Il n'a aucun effet sous Mac OS.

**Type**

Booléen ; lecture/écriture.

**Attribut Application exitCode**`app.exitCode`**Description**

Code numérique d'état utilisé lors de l'exécution d'un script en externe (c'est-à-dire à partir d'une ligne de commande ou d'AppleScript).

- Sous Windows, lorsque After Effects est lancé à partir de la ligne de commande (à l'aide de la commande `afterfx` ou `afterfx -m`) et qu'un script est spécifié à l'aide de l'option `-r` ou `-s`, la valeur est renvoyée sur la ligne de commande.
- Sous Mac OS, la valeur est renvoyée en tant que résultat de la commande AppleScript `DoScript` pour chaque script.

Sous Mac OS comme sous Windows, la valeur est définie sur 0 (`EXIT_SUCCESS`) au début de chaque évaluation de script. En cas d'erreur lors de l'exécution du script, celui-ci peut définir cette valeur sur un nombre entier positif indiquant l'erreur qui s'est produite.

**Type**

Entier ; lecture/écriture.

**Exemple**

```
app.exitCode = 2; //on quit, if value is 2, an error has occurred
```

**Attribut Application isRenderEngine**`app.isRenderEngine`**Description**

Cet attribut a la valeur `true` si l'application After Effects est exécutée en tant que moteur de rendu.

**Type**

Booléen ; lecture seule.

**Attribut Application isWatchFolder**`app.isWatchFolder`**Description**

Cet attribut a la valeur `true` si la boîte de dialogue Dossier de contrôle est affichée actuellement et si l'application est en train de contrôler un dossier pour le rendu.

**Type**

Booléen ; lecture seule.

**Attribut Application language**`app.language`**Description**

Langue dans laquelle After Effects est exécuté.

**Type**

Valeur énumérée `Language` ; lecture seule. L'une des suivantes :

```
Language.ENGLISH  
Language.FRENCH  
Language.GERMAN  
Language.ITALIAN  
Language.JAPANESE  
Language.SPANISH
```

**Exemple**

```
var lang = app.language;  
if (lang == Language.ENGLISH)  
    alert("After Effects is running in English.");  
else if (lang == Language.FRENCH)  
    alert("After Effects is running in French.");  
else  
    alert("After Effects is not running in English or French.");
```

**Attribut `Application memoryInUse`**

```
app.memoryInUse
```

**Description**

Nombre d'octets de mémoire actuellement utilisés par cette application.

**Type**

Nombre ; lecture seule.

**Méthode `Application newProject()`**

```
app.newProject()
```

**Description**

Crée un nouveau projet dans After Effects, en reproduisant la commande de menu Fichier > Créer > Nouveau projet.

Si le projet en cours a été modifié, l'utilisateur est invité à l'enregistrer. Si l'utilisateur ferme la boîte de dialogue Enregistrer sans enregistrer, le nouveau projet n'est pas créé et la méthode renvoie une valeur nulle. Utilisez `app.project.close(CloseOptions.DO_NOT_SAVE_CHANGES)` pour fermer le projet en cours avant d'en ouvrir un nouveau. Voir la section « Méthode `Project close()` », page 112.

**Paramètres**

Aucun.

**Renvoi**

Un nouvel objet `Project` ou une valeur nulle si aucun nouveau projet n'est créé.

**Exemple**

```
app.project.close(CloseOptions.DO_NOT_SAVE_CHANGES);  
app.newProject();
```

## Attribut Application onError

app.onError

### Description

Nom d'une fonction de rappel qui est appelée lorsqu'une erreur se produit. En créant une fonction et en l'attribuant à cet attribut, il est possible de réagir aux erreurs systématiquement. Vous pouvez, par exemple, fermer et redémarrer l'application, et noter l'erreur dans un fichier journal si elle est survenue en cours de rendu. Voir la section « Méthode RenderQueue render() », page 154.

La chaîne d'erreur et une chaîne de gravité sont transmises à la fonction de rappel. Aucune valeur ne doit être renvoyée.

### Type

Chaîne de nom de fonction ou valeur nulle si aucune fonction n'est attribuée ; lecture/écriture.

### Exemple

```
function err(errString) {  
  alert(errString);  
}  
app.onError = err;
```

## Méthode Application open()

app.open()  
app.open(*file*)

### Description

Ouvre un projet.

### Paramètres

file	Facultatif. Objet ExtendScript File pour le fichier de projet à ouvrir. S'il n'est pas fourni, la méthode invite l'utilisateur à sélectionner un fichier de projet.
------	---

### Renvoi

Un nouvel objet Project pour le projet spécifié ou une valeur nulle si l'utilisateur ferme la boîte de dialogue Ouvrir sans effectuer de sélection.

### Exemple

```
var my_file = new File("../my_folder/my_test.aep");  
if (my_file.exists){  
  new_project = app.open(my_file);  
  if (new_project){  
    alert(new_project.file.name);  
  }  
}
```

**Méthode Application parseSwatchFile()**app.parseSwatchFile(*file*)**Description**

Charge des données d'échantillon de couleur à partir d'un fichier Adobe Swatch Exchange (ASE).

**Paramètres**

file	Spécification du fichier, qui est un objet ExtendScript File.
------	---

**Renvoi**

Les données de l'échantillon, au format suivant :

data.majorVersion data.minorVersion	Numéro de version ASE.
data.values	Tableau de valeurs SwatchValue.
SwatchValue.type	L'un des types suivants : RGB, CMYK, LAB, Gray
SwatchValue.r SwatchValue.g SwatchValue.b	Lorsque type = "RGB", les valeurs de couleur comprises dans la plage [0.0..1.0]. 0, 0, 0 correspond à la valeur Black.
SwatchValue.c SwatchValue.m SwatchValue.y SwatchValue.k	Lorsque type = "CMYK", les valeurs de couleur comprises dans la plage [0.0..1.0]. 0, 0, 0, 0 correspond à la valeur White.
SwatchValue.L SwatchValue.a SwatchValue.b	Lorsque type = "LAB", les valeurs de couleur. L est compris dans la plage [0.0..1.0]. a et b sont compris dans la plage [-128.0..+128.0] 0, 0, 0 correspond à la valeur Black.
SwatchValue.value	Lorsque type = "Gray", value est compris dans la plage [0.0..1.0]. 0.0 correspond à la valeur Black.

**Méthode Application pauseWatchFolder()**app.pauseWatchFolder(*pause*)**Description**

Interrompt temporairement ou reprend la recherche d'éléments de rendu dans le dossier de contrôle cible.

**Paramètres**

pause	True permet d'interrompre et false de reprendre.
-------	--

**Renvoi**

Aucun.

**Voir aussi**

- « Attribut Application isWatchFolder », page 24
- « Méthode Application watchFolder() », page 30
- « Méthode Application endWatchFolder() », page 23

**Attribut Application project**

app.project

**Description**

Projet en cours. Voir la section « Objet Project », page 110.

**Type**

Objet Project ; lecture seule.

**Méthode Application purge()**

app.purge(*target*)

**Description**

Purge de la mémoire les données inutilisées des types spécifiés. Reproduit les options de la commande Purger du menu Edition.

**Paramètres**

target	<p>Type d'élément à purger de la mémoire ; valeur énumérée PurgeTarget, l'une des suivantes :</p> <ul style="list-style-type: none"> <li>• PurgeTarget.ALL_CACHES : purge toutes les données qu'After Effects a mises en mémoire cache dans la mémoire physique.</li> <li>• PurgeTarget.UNDO_CACHES : purge toutes les données enregistrées dans le cache d'annulation.</li> <li>• PurgeTarget.SNAPSHOT_CACHES : purge toutes les données mises en cache en tant qu'instantanés de composition/calque.</li> <li>• PurgeTarget.IMAGE_CACHES : purge toutes les données d'image enregistrées.</li> </ul>
--------	--

**Renvoi**

Aucun.

**Méthode Application quit()**

app.quit()

**Description**

Ferme l'application After Effects.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Attribut Application saveProjectOnCrash**`app.saveProjectOnCrash`**Description**

Lorsque cet attribut est défini sur `true` (valeur par défaut), After Effects tente d'afficher une boîte de dialogue qui vous permet d'enregistrer le projet en cours lorsqu'une erreur provoquant la fermeture inopinée de l'application se produit. Lorsqu'il est défini sur `false`, cette boîte de dialogue disparaît et l'application se ferme sans que le projet ne soit enregistré.

**Type**

Booléen ; lecture/écriture.

**Méthode Application scheduleTask()**`app.scheduleTask(stringToExecute, delay, repeat)`**Description**

Planifie le JavaScript spécifié pour une exécution ultérieure.

**Paramètres**

<code>stringToExecute</code>	Chaîne contenant le JavaScript à exécuter.
<code>delay</code>	Durée d'attente (en millisecondes) avant l'exécution du JavaScript. Valeur à virgule flottante.
<code>repeat</code>	Lorsque ce paramètre est défini sur <code>true</code> , répète l'exécution du script, en respectant le délai spécifié entre chaque exécution. Lorsqu'il est défini sur <code>false</code> , le script n'est exécuté qu'une seule fois.

**Renvoi**

Nombre entier, identificateur unique de cette tâche, pouvant permettre d'annuler la tâche via la méthode `app.cancelTask()`.

**Méthode Application setMemoryUsageLimits()**`app.setMemoryUsageLimits(imageCachePercentage, maximumMemoryPercentage)`**Description**

Définit les limites d'utilisation de mémoire, telles que spécifiées dans les préférences Mémoire et cache. Pour les deux valeurs, si la mémoire vive installée est inférieure à une quantité donnée ( $n$  giga-octets), la valeur correspond à un pourcentage de la mémoire vive installée ; sinon, elle correspond à un pourcentage de  $n$ . La valeur de  $n$  est : 2 Go pour Windows 32 bits, 4 Go pour Windows 64 bits, 3,5 Go pour Mac OS.

**Paramètres**

<code>imageCachePercentage</code>	Valeur à virgule flottante ; pourcentage de mémoire assigné au cache image.
<code>maximumMemoryPercentage</code>	Valeur à virgule flottante ; pourcentage d'utilisation maximale de mémoire.

**Renvoi**

Aucun.

**Méthode Application `setSavePreferencesOnQuit()`**

```
app.setSavePreferencesOnQuit(doSave)
```

**Description**

Définit ou supprime l'indicateur qui détermine si les préférences doivent être enregistrées à la fermeture de l'application.

**Paramètres**

<code>doSave</code>	Lorsque ce paramètre est défini sur <code>true</code> , les préférences sont enregistrées à la fermeture de l'application. Lorsqu'il est défini sur <code>false</code> , elles ne sont pas enregistrées.
---------------------	--

**Renvoi**

Aucun.

**Attribut Application `settings`**

```
app.settings
```

**Description**

Paramètres en cours. Voir la section « `Objet Settings` », page 164.

**Type**

Objet `Settings` ; lecture seule.

**Attribut Application `version`**

```
app.version
```

**Description**

Chaîne alphanumérique indiquant la version d'After Effects en cours d'exécution.

**Type**

Chaîne ; lecture seule.

**Exemple**

```
var ver = app.version;
alert("This machine is running version " + ver + " of After Effects.");
```

**Méthode Application `watchFolder()`**

```
app.watchFolder(folder_object_to_watch)
```

**Description**

Démarre un processus de dossier de contrôle (rendu en réseau) pointant vers un dossier spécifié.

**Paramètres**

<code>folder_object_to_watch</code>	Objet <code>ExtendScript Folder</code> que le dossier doit contrôler.
-------------------------------------	---

**Renvoi**

Aucun.

**Exemple**

```
var theFolder = new Folder("c:/tool");  
app.watchFolder(theFolder);
```

**Voir aussi**

« Méthode Application endWatchFolder() », page 23

« Méthode Application parseSwatchFile() », page 27

« Attribut Application isWatchFolder », page 24

## Objet AVItem

app.project.item(*index*)

### Description

L'objet AVItem fournit l'accès aux attributs et aux méthodes des fichiers audiovisuels importés dans After Effects.

- AVItem est une sous-classe d'Item. Toutes les méthodes et tous les attributs d'Item, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez AVItem. Voir la section « Objet Item », page 76.
- AVItem est la classe de base de CompItem et de FootageItem, de sorte que les attributs et les méthodes AVItem sont également disponibles lors de l'utilisation des objets CompItem et FootageItem. Voir les sections « Objet CompItem », page 52 et « Objet FootageItem », page 64.

### Attributs

Attribut	Référence	Description
name	« Attribut AVItem name », page 35	Nom de l'objet, tel qu'affiché dans le panneau Projet.
width	« Attribut AVItem width », page 38	Largeur de l'élément.
height	« Attribut AVItem height », page 34	Hauteur de l'élément.
pixelAspect	« Attribut AVItem pixelAspect », page 35	Format des pixels de l'élément.
frameRate	« Attribut AVItem frameRate », page 34	Cadence d'image de l'élément.
frameDuration	« Attribut AVItem frameDuration », page 33	Durée d'image de l'élément.
duration	« Attribut AVItem duration », page 33	Durée totale de l'élément.
useProxy	« Attribut AVItem useProxy », page 38	Lorsque cet attribut est défini sur true, une source de doublure est utilisée pour cet élément.
proxySource	« Attribut AVItem proxySource », page 35	Objet FootageItem utilisé en tant que doublure pour l'élément.
time	« Attribut AVItem time », page 38	Instant actuel de l'élément.
usedIn	« Attribut AVItem usedIn », page 38	Objets CompItem qui utilisent cet élément.
hasVideo	« Attribut AVItem hasVideo », page 34	Lorsque cet attribut est défini sur true, l'élément possède un composant vidéo.
hasAudio	« Attribut AVItem hasAudio », page 34	Lorsque cet attribut est défini sur true, l'élément possède un composant audio.
footageMissing	« Attribut AVItem footageMissing », page 33	Lorsque cet attribut est défini sur true, l'élément est introuvable ou il s'agit d'un pseudo-élément.

### Méthodes

Méthode	Référence	Description
setProxy()	« Méthode AVItem setProxy() », page 36	Définit une doublure pour l'élément.
setProxyWithSequence()	« Méthode AVItem setProxyWithSequence() », page 37	Définit une séquence comme doublure pour l'élément.
setProxyWithSolid()	« Méthode AVItem setProxyWithSolid() », page 37	Définit un solide comme doublure pour l'élément.
setProxyWithPlaceholder()	« Méthode AVItem setProxyWithPlaceholder() », page 36	Définit un pseudo-élément comme doublure pour l'élément.
setProxyToNone()	« Méthode AVItem setProxyToNone() », page 36	Supprime la doublure pour l'élément.

**Attribut AVItem duration**

`app.project.item(index).duration`

**Description**

Renvoie la durée, en secondes, de l'élément. Les éléments de métrage fixe ont une durée de 0.

- Dans un objet `CompItem`, la valeur est liée à la durée de la composition. Elle est accessible en lecture/écriture.
- Dans un objet `FootageItem`, la valeur est liée à la durée de l'objet `mainSource`. Elle est accessible en lecture seule.

**Type**

Valeur à virgule flottante comprise dans la plage [0.0..10800.0] ; lecture/écriture pour un objet `CompItem` ; lecture seule dans les autres cas.

**Attribut AVItem footageMissing**

`app.project.item(index).footageMissing`

**Description**

Lorsque cet attribut est défini sur `true`, l'`AVItem` est un pseudo-élément ou représente un métrage avec un fichier source introuvable. Dans ce cas, le chemin du fichier source manquant se trouve dans l'attribut `missingFootagePath` de l'objet du fichier source de l'élément de métrage. Voir les sections « Attribut `FootageItem mainSource` », page 65 et « Attribut `FileSource missingFootagePath` », page 60.

**Type**

Booléen ; lecture seule.

**Attribut AVItem frameDuration**

`app.project.item(index).frameDuration`

**Description**

Renvoie la longueur, en secondes, d'une image pour cet `AVItem`. Réciproque de `frameRate`. Lorsque cet attribut est défini, la réciproque est automatiquement définie comme une nouvelle valeur de `frameRate`.

Cet attribut renvoie la réciproque de `frameRate`, qui est susceptible d'être différente d'une valeur que vous avez définie, si cette valeur n'est pas divisible régulièrement par 1,0 (par exemple, 0,3). En raison de limitations numériques,  $(1 / (1 / 0,3))$  est proche de 0,3 mais n'est pas égal à 0,3.

Si l'`AVItem` est un `FootageItem`, cette valeur est liée à l'objet `mainSource`, et est en lecture seule. Pour la modifier, définissez l'attribut `conformFrameRate` de l'objet `mainSource`. Cela permet de définir les valeurs `frameRate` et `frameDuration` du `FootageItem`.

**Type**

Valeur à virgule flottante comprise dans la plage [1/99.. 1.0] ; lecture seule pour un `FootageItem`, lecture/écriture dans les autres cas.

**Attribut AVItem frameRate**

`app.project.item(index).frameRate`

**Description**

Cadence d'image de l'AVItem, en nombre d'images par seconde. Réciproque de l'attribut `frameDuration`. Lorsque cet attribut est défini, sa réciproque est automatiquement définie comme une nouvelle valeur `frameDuration`.

- Dans un `CompItem`, la valeur est liée à l'attribut `frameRate` de la composition et est en lecture/écriture.
- Dans un `FootageItem`, la valeur est liée à l'attribut `frameRate` de l'objet `mainSource` et est en lecture seule. Pour la modifier, définissez l'attribut `conformFrameRate` de l'objet `mainSource`. Cela permet de définir les valeurs `frameRate` et `frameDuration` du `FootageItem`.

**Type**

Valeur à virgule flottante comprise dans la plage [1.0..99.0] ; lecture seule pour un `FootageItem`, lecture/écriture dans les autres cas.

**Attribut AVItem hasAudio**

`app.project.item(index).hasAudio`

**Description**

Lorsque cet attribut a la valeur `true`, l'AVItem possède un composant audio.

- Dans un `CompItem`, la valeur est liée à la composition.
- Dans un `FootageItem`, la valeur est liée à l'objet `mainSource`.

**Type**

Booléen ; lecture seule.

**Attribut AVItem hasVideo**

`app.project.item(index).hasVideo`

**Description**

Lorsque cet attribut a la valeur `true`, l'AVItem possède un composant audio.

- Dans un `CompItem`, la valeur est liée à la composition.
- Dans un `FootageItem`, la valeur est liée à l'objet `mainSource`.

**Type**

Booléen ; lecture seule.

**Attribut AVItem height**

`app.project.item(index).height`

**Description**

Hauteur de l'élément, en pixels.

- Dans un `CompItem`, la valeur est liée à la composition. Elle est en lecture/écriture.
- Dans un `FootageItem`, la valeur est liée à l'objet `mainSource`. Elle est en lecture/écriture uniquement si l'objet `mainSource` est un `SolidSource`. Dans les autres cas, elle est en lecture seule.

**Type**

Nombre entier compris dans la plage de valeurs [1...30000] ; lecture/écriture, sauf dans les cas indiqués.

**Attribut AVItem name**`app.project.item(index).name`**Description**

Nom de l'élément, tel qu'affiché dans le panneau Projet.

- Dans un FootageItem, la valeur est liée à l'objet mainSource. Si l'objet mainSource est un FileSource, cette valeur contrôle le nom d'affichage qui apparaît dans le panneau Projet, mais n'affecte pas le nom de fichier.

**Type**

Chaîne ; lecture/écriture.

**Attribut AVItem pixelAspect**`app.project.item(index).pixelAspect`**Description**

Format des pixels de l'élément.

- Dans un CompItem, la valeur est liée à la composition.
- Dans un FootageItem, la valeur est liée à l'objet mainSource.

Certaines valeurs pixelAspect sont connues d'After Effects et sont stockées/extraites avec précision. Ces valeurs sont celles de la série {1, 0.9, 1.2, 1.07, 1.42, 2, 0.95, 1.9}. Les autres valeurs peuvent présenter de légères erreurs d'arrondi lorsque vous les définissez ou que vous les extrayez. Par conséquent, la valeur que vous extrayez après la définition peut être légèrement différente de celle que vous avez fournie.

**Type**

Valeur à virgule flottante comprise dans la plage [0.01..100.0] ; lecture/écriture.

**Attribut AVItem proxySource**`app.project.item(index).proxySource`**Description**

FootageSource utilisé comme doublure. L'attribut est en lecture seule. Pour le modifier, appelez l'une des méthodes AVItem qui modifient la source de la doublure : setProxy(), setProxyWithSequence(), setProxyWithSolid() ou setProxyWithPlaceholder().

**Type**

Objet FootageSource ; lecture seule.

**Méthode AVItem setProxy()**

```
app.project.item(index).setProxy(file)
```

**Description**

Définit un fichier comme doublure de cet AVItem. Charge le fichier spécifié dans un nouvel objet FileSource, définit celui-ci comme la valeur de l'attribut `proxySource` et définit `useProxy` sur `true`. Cette méthode ne conserve pas les paramètres d'interprétation mais utilise les préférences utilisateur. Si le fichier comporte une couche alpha non identifiée et si la préférence utilisateur spécifie d'afficher un message demandant à l'utilisateur ce qu'il souhaite faire, la méthode évalue l'interprétation alpha au lieu de la demander à l'utilisateur.

Cette action est différente de ce qui se produit lorsque vous définissez la source principale d'un FootageItem, mais ces deux actions s'effectuent comme dans l'interface utilisateur.

**Paramètres**

file	Objet ExtendScript File pour le fichier à utiliser en tant que doublure.
------	--

**Renvoi**

Aucun.

**Méthode AVItem setProxyToNone()**

```
app.project.item(index).setProxyToNone()
```

**Description**

Supprime la doublure de cet AVItem, définit `proxySource` sur une valeur nulle et `useProxy` sur `false`.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Méthode AVItem setProxyWithPlaceholder()**

```
app.project.item(index).setProxyWithPlaceholder(name, width, height, frameRate, duration)
```

**Description**

Crée un objet PlaceholderSource avec les valeurs spécifiées, définit celui-ci en tant que valeur de l'attribut `proxySource` et définit `useProxy` sur `true`. Cette méthode ne conserve pas les paramètres d'interprétation mais utilise les préférences utilisateur.

*REMARQUE : il n'existe pas de moyen direct de définir un pseudo-élément en tant que doublure dans l'interface utilisateur ; ce comportement se produit lorsqu'une doublure a été définie, puis déplacée ou supprimée.*

**Paramètres**

name	Chaîne contenant le nom du nouvel objet.
width, height	Dimensions du pseudo-élément, en pixels. Nombre entier compris dans la plage [4..30000].
frameRate	Nombre d'images par seconde. Nombre entier compris dans la plage [1..99].
duration	Durée totale, en secondes (3 heures max.). Nombre entier compris dans la plage [0.0..10800.0].

**Renvoi**

Aucun.

**Méthode AVItem setProxyWithSequence()**

```
app.project.item(index).setProxyWithSequence(file, forceAlphabetical)
```

**Description**

Définit une séquence de fichiers en tant que doublure de cet AVItem, avec l'option permettant de forcer l'ordre alphabétique. Charge la séquence de fichiers spécifiée dans un nouvel objet FileSource, définit celui-ci comme la valeur de l'attribut proxySource et définit useProxy sur true. Cette méthode ne conserve pas les paramètres d'interprétation mais utilise les préférences utilisateur. Si un fichier comporte une couche alpha non identifiée et si la préférence utilisateur spécifie d'afficher un message demandant à l'utilisateur ce qu'il souhaite faire, la méthode évalue l'interprétation alpha au lieu de la demander à l'utilisateur.

**Paramètres**

file	Objet ExtendScript File pour le premier fichier de la séquence.
forceAlphabetical	Lorsque ce paramètre est true, l'option Ordre alphabétique doit être utilisée.

**Renvoi**

Aucun.

**Méthode AVItem setProxyWithSolid()**

```
app.project.item(index).setProxyWithSolid(color, name, width, height, pixelAspect)
```

**Description**

Crée un objet SolidSource avec les valeurs spécifiées, définit celui-ci en tant que valeur de l'attribut proxySource et définit useProxy sur true. Cette méthode ne conserve pas les paramètres d'interprétation mais utilise les préférences utilisateur.

*REMARQUE : l'interface utilisateur n'offre aucun moyen de définir un solide comme doublure ; cette fonction n'est disponible que par le biais de la création de script.*

**Paramètres**

color	Couleur du solide, tableau de trois valeurs à virgule flottante, [R, V, B], comprises dans la plage [0.0..1.0].
name	Chaîne contenant le nom du nouvel objet.
width, height	Dimensions du pseudo-élément, en pixels. Nombre entier compris dans la plage [1...30000].
pixelAspect	Format des pixels du solide. Valeur à virgule flottante comprise dans la plage [0.01... 100.0].

**Renvoi**

Aucun.

**Attribut AVItem time**

`app.project.item(index).time`

**Description**

Instant courant de l'élément lorsqu'il est prévisualisé directement à partir du panneau Projet. Cette valeur correspond à un nombre de secondes. Utilisez la méthode globale `timeToCurrentFormat` pour la convertir en une valeur de type chaîne exprimant l'instant en termes d'images. Voir la section « Fonction globale `timeToCurrentFormat()` », page 17.

Ne définissez pas cette valeur pour un `FootageItem` dont la valeur `mainSource` est fixe (`item.mainSource.isStill` est `true`).

**Type**

Valeur à virgule flottante ; lecture/écriture.

**Attribut AVItem usedIn**

`app.project.item(index).usedIn`

**Description**

Toutes les compositions qui utilisent cet AVItem.

Veillez noter que la valeur du tableau est copiée suite à une extraction. Par conséquent, elle n'est pas automatiquement mise à jour. Si vous récupérez cette valeur, puis que vous l'ajoutez dans une autre composition, vous devez de nouveau extraire la valeur pour obtenir un tableau incluant le nouvel élément.

**Type**

Tableau d'objets `CompItem` ; lecture seule.

**Attribut AVItem useProxy**

`app.project.item(index).useProxy`

**Description**

Lorsque cet attribut est `true`, une doublure est utilisée pour l'élément. Il est défini sur `true` par toutes les méthodes `SetProxy` et sur `false` par la méthode `SetProxyToNone()`.

**Type**

Booléen ; lecture/écriture.

**Attribut AVItem width**

`app.project.item(index).width`

**Description**

Largeur de l'élément, en pixels.

- Dans un `CompItem`, la valeur est liée à la composition. Elle est en lecture/écriture.
- Dans un `FootageItem`, la valeur est liée à l'objet `mainSource`. Elle est en lecture/écriture uniquement si l'objet `mainSource` est un `SolidSource`. Dans les autres cas, elle est en lecture seule.

**Type**

Nombre entier compris dans la plage de valeurs [1...30000] ; lecture/écriture, sauf dans les cas indiqués.

## Objet AVLayer

`app.project.item(index).layer(index)`

### Description

L'objet AVLayer fournit une interface aux calques contenant des objets AVItem (calques de composition, de métrage, de solide, de texte et de son).

- AVLayer est une sous-classe de Layer. Toutes les méthodes et tous les attributs de Layer, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez AVLayer. Voir la section « Objet Layer », page 83.
- AVLayer est une classe de base de TextLayer. Ainsi, les attributs et les méthodes AVLayer sont disponibles lorsque vous utilisez des objets TextLayer. Voir la section « Objet TextLayer », page 174.

### Propriétés AE

Les différents types de calque ont des propriétés AE différentes. AVLayer possède les propriétés et groupes de propriétés suivants :

Marker

Time Remap

Motion Trackers

Masks

Effects

Transform

  Anchor Point

  Position

  Scale

  Orientation

  X Rotation

  Y Rotation

  Rotation

  Opacity

Layer Styles

Material Options

  Casts Shadows

  Light Transmission

  Accepts Shadows

  Accepts Lights

  Ambient

  Diffuse

  Specular

  Shininess

  Metal

Audio

  Audio Levels

### Exemple

Si le premier élément du projet est un CompItem et si le premier calque de ce CompItem est un AVLayer, le code suivant définit la qualité (quality) du calque, son instant de début (startTime) et son point d'entrée (inPoint).

```
var firstLayer = app.project.item(1).layer(1);
firstLayer.quality = LayerQuality.BEST;
firstLayer.startTime = 1;
firstLayer.inPoint = 2;
```

**Attributs**

Attribut	Référence	Description
source	« Attribut AVLayer source », page 47	Élément source pour ce calque.
isNameFromSource	« Attribut AVLayer isNameFromSource », page 46	Lorsque cet attribut est true, le calque ne possède pas de nom explicitement défini, mais contient une source nommée.
height	« Attribut AVLayer height », page 46	Hauteur du calque.
width	« Attribut AVLayer width », page 49	Largeur du calque.
audioEnabled	« Attribut AVLayer audioEnabled », page 42	Lorsque cet attribut est true, l'audio du calque est activé.
motionBlur	« Attribut AVLayer motionBlur », page 46	Lorsque cet attribut est true, le flou directionnel du calque est activé.
effectsActive	« Attribut AVLayer effectsActive », page 44	Lorsque cet attribut est true, les effets du calque sont actifs.
adjustmentLayer	« Attribut AVLayer adjustmentLayer », page 41	Lorsque cet attribut est true, il s'agit d'un calque d'effets.
guideLayer	« Attribut AVLayer guideLayer », page 45	Lorsque cet attribut est true, il s'agit d'un calque repère.
threeDLayer	« Attribut AVLayer threeDLayer », page 48	Lorsque cet attribut est true, il s'agit d'un calque 3D.
threeDPerChar	« Attribut AVLayer threeDPerChar », page 48	Lorsque cet attribut a la valeur true, la projection 3D est définie caractère par caractère sur ce calque de texte.
canSetCollapseTransformation	« Attribut AVLayer canSetCollapseTransformation », page 44	Lorsque cet attribut est true, il est permis de modifier la valeur de collapseTransformation.
collapseTransformation	« Attribut AVLayer collapseTransformation », page 44	Lorsque cet attribut est true, la condensation des transformations est activée.
frameBlending	« Attribut AVLayer frameBlending », page 45	Lorsque cet attribut est true, l'interpolation est activée.
frameBlendingType	« Attribut AVLayer frameBlendingType », page 45	Il s'agit du type d'interpolation appliqué au calque.
canSetTimeRemapEnabled	« Attribut AVLayer canSetTimeRemapEnabled », page 44	Lorsque cet attribut est true, il est permis de modifier la valeur de timeRemapEnabled.
timeRemapEnabled	« Attribut AVLayer timeRemapEnabled », page 48	Lorsque cet attribut est true, le remappage temporel est activé pour ce calque.
hasAudio	« Attribut AVLayer hasAudio », page 45	Lorsque cet attribut est true, le calque contient un composant audio.
audioActive	« Attribut AVLayer audioActive », page 41	Lorsque cet attribut est true, le composant audio du calque est actif à l'instant courant.
blendingMode	« Attribut AVLayer blendingMode », page 42	Mode de fusion du calque.
preserveTransparency	« Attribut AVLayer preserveTransparency », page 46	Lorsque cet attribut est true, l'option Préserver la transparence est activée.
trackMatteType	« Attribut AVLayer trackMatteType », page 49	Si le calque comporte un cache par approche, cet attribut indique la manière dont il est appliqué.
isTrackMatte	« Attribut AVLayer isTrackMatte », page 46	Lorsque cet attribut est true, le calque est utilisé comme cache par approche pour le calque sous-jacent.

Attribut	Référence	Description
hasTrackMatte	« Attribut AVLayer hasTrackMatte », page 45	Lorsque cet attribut est true, le calque immédiatement supérieur est utilisé comme cache par approche sur ce calque.
quality	« Attribut AVLayer quality », page 47	Paramètre de qualité du calque.
autoOrient	« Attribut AVLayer autoOrient », page 42	Il s'agit du type d'orientation automatique appliqué au calque.

### Méthodes

Méthode	Référence	Description
audioActiveAtTime()	« Méthode AVLayer audioActiveAtTime() », page 42	Signale si le composant audio de ce calque est actif à un instant donné.
calculateTransformFromPoints()	« Méthode AVLayer calculateTransformFromPoints() », page 43	Calcule une transformation à partir d'un ensemble de points de ce calque.
replaceSource()	« Méthode AVLayer replaceSource() », page 47	Modifie l'élément source pour ce calque.
sourceRectAtTime()	« Méthode AVLayer sourceRectAtTime() », page 47	Extrait le rectangle source d'un calque.

### Attribut AVLayer adjustmentLayer

`app.project.item(index).layer(index).adjustmentLayer`

#### Description

True, s'il s'agit d'un calque d'effets.

#### Type

Booléen ; lecture/écriture.

### Attribut AVLayer audioActive

`app.project.item(index).layer(index).audioActive`

#### Description

True si le composant audio du calque est actif à l'instant courant.

Pour que la valeur soit true, `audioEnabled` doit également être true, aucun autre calque comportant de l'audio ne doit être isolé sauf si ce calque l'est aussi, et l'instant donné doit être compris entre le point d'entrée et le point de sortie de ce calque.

#### Type

Booléen ; lecture seule.

**Méthode AVLayer audioActiveAtTime()**

```
app.project.item(index).layer(index).audioActiveAtTime(time)
```

**Description**

Renvoie true si le composant audio de ce calque est actif à l'instant spécifié.

Pour que cette méthode renvoie true, `audioEnabled` doit également être true, aucun autre calque comportant de l'audio ne doit être isolé sauf si ce calque l'est aussi, et l'instant donné doit être compris entre le point d'entrée et le point de sortie de ce calque.

**Paramètres**

time	L'instant, en secondes. Valeur à virgule flottante.
------	---

**Renvoi**

Booléen.

**Attribut AVLayer audioEnabled**

```
app.project.item(index).layer(index).audioEnabled
```

**Description**

Lorsque cet attribut est true, l'audio du calque est activé. Cette valeur correspond à la commande d'activation/désactivation du composant audio dans le panneau Montage.

**Type**

Booléen ; lecture/écriture.

**Attribut AVLayer autoOrient**

```
app.project.item(index).layer(index).autoOrient
```

**Description**

Il s'agit du type d'orientation automatique à appliquer au calque.

**Type**

Valeur énumérée `AutoOrientType` ; lecture/écriture. L'une des suivantes :

`AutoOrientType.ALONG_PATH`

`AutoOrientType.CAMERA_OR_POINT_OF_INTEREST`

`AutoOrientType.NO_AUTO_ORIENT`

**Attribut AVLayer blendingMode**

```
app.project.item(index).layer(index).blendingMode
```

**Description**

Mode de fusion du calque.

**Type**

Valeur énumérée `BlendingMode` ; lecture/écriture. L'une des suivantes :

`BlendingMode.ADD`  
`BlendingMode.ALPHA_ADD`  
`BlendingMode.CLASSIC_COLOR_BURN`  
`BlendingMode.CLASSIC_COLOR_DODGE`  
`BlendingMode.CLASSIC_DIFFERENCE`  
`BlendingMode.COLOR`  
`BlendingMode.COLOR_BURN`  
`BlendingMode.COLOR_DODGE`  
`BlendingMode.DANCING DISSOLVE`  
`BlendingMode.DARKEN`  
`BlendingMode.DARKER_COLOR`  
`BlendingMode.DIFFERENCE`  
`BlendingMode.DISSOLVE`  
`BlendingMode.EXCLUSION`  
`BlendingMode.HARD_LIGHT`  
`BlendingMode.HARD_MIX`  
`BlendingMode.HUE`  
`BlendingMode.LIGHTEN`  
`BlendingMode.LIGHTER_COLOR`  
`BlendingMode.LINEAR_BURN`  
`BlendingMode.LINEAR_DODGE`  
`BlendingMode.LINEAR_LIGHT`  
`BlendingMode.LUMINESCENT_PREMUL`  
`BlendingMode.LUMINOSITY`  
`BlendingMode.MULTIPLY`  
`BlendingMode.NORMAL`  
`BlendingMode.OVERLAY`  
`BlendingMode.PIN_LIGHT`  
`BlendingMode.SATURATION`  
`BlendingMode.SCREEN`  
`BlendingMode.SILHOUETE_ALPHA`  
`BlendingMode.SILHOUETTE_LUMA`  
`BlendingMode.SOFT_LIGHT`  
`BlendingMode.STENCIL_ALPHA`  
`BlendingMode.STENCIL_LUMA`  
`BlendingMode.VIVID_LIGHT`

**Méthode AVLayer calculateTransformFromPoints()**

`app.project.item(index).layer(index).calculateTransformFromPoints(pointTopLeft, pointTopRight, pointBottomRight)`

**Description**

Calcule une transformation à partir d'un ensemble de points de ce calque.

**Paramètres**

<code>pointTopLeft</code>	Coordonnées du point supérieur gauche sous forme de tableau, [x, y, z].
<code>pointTopRight</code>	Coordonnées du point supérieur droit sous forme de tableau, [x, y, z].
<code>pointBottomRight</code>	Coordonnées du point inférieur droit sous forme de tableau, [x, y, z].

**Renvoi**

Un objet avec les propriétés de transformation définies.

**Exemple**

```
var newLayer = comp.layers.add(newFootage);
newLayer.threeDLayer = true;

newLayer.blendingMode = BlendingMode.ALPHA_ADD;
var transform = newLayer.calculateTransformFromPoints(tl, tr, bl);
for(var sel in transform) {
  newLayer.transform[sel].setValue(transform[sel]);
}
```

**Attribut AVLayer canSetCollapseTransformation**

```
app.project.item(index).layer(index).canSetCollapseTransformation
```

**Description**

True s'il est permis de modifier la valeur de l'attribut collapseTransformation sur ce calque.

**Type**

Booléen ; lecture seule.

**Attribut AVLayer canSetTimeRemapEnabled**

```
app.project.item(index).layer(index).canSetTimeRemapEnabled
```

**Description**

True s'il est permis de modifier la valeur de l'attribut timeRemapEnabled sur ce calque.

**Type**

Booléen ; lecture seule.

**Attribut AVLayer collapseTransformation**

```
app.project.item(index).layer(index).collapseTransformation
```

**Description**

True si la condensation des transformations est activée pour ce calque.

**Type**

Booléen ; lecture/écriture.

**Attribut AVLayer effectsActive**

```
app.project.item(index).layer(index).effectsActive
```

**Description**

True si les effets du calque sont activés, comme l'indique l'icône <f> dans l'interface utilisateur.

**Type**

Booléen ; lecture/écriture.

**Attribut AVLayer frameBlending**

`app.project.item(index).layer(index).frameBlending`

**Description**

True si l'interpolation est activée pour ce calque.

**Type**

Booléen ; lecture seule.

**Attribut AVLayer frameBlendingType**

`app.project.item(index).layer(index).frameBlendingType`

**Description**

Type d'interpolation à effectuer lorsque l'interpolation est activée pour le calque.

**Type**

Valeur énumérée `FrameBlendingType` ; lecture/écriture. L'une des suivantes :

`FrameBlendingType.FRAME_MIX`

`FrameBlendingType.NO_FRAME_BLEND`

`FrameBlendingType.PIXEL_MOTION`

**Attribut AVLayer guideLayer**

`app.project.item(index).layer(index).guideLayer`

**Description**

True si le calque est un calque repère.

**Type**

Booléen ; lecture/écriture.

**Attribut AVLayer hasAudio**

`app.project.item(index).layer(index).hasAudio`

**Description**

True si le calque contient un composant audio, que ce composant audio soit activé ou isolé (option Solo).

**Type**

Booléen ; lecture seule.

**Attribut AVLayer hasTrackMatte**

`app.project.item(index).layer(index).hasTrackMatte`

**Description**

True si le calque supérieur à ce calque est utilisé comme cache par approche sur ce calque. Lorsque cet attribut est true, la valeur `trackMatteType` de ce calque définit la façon dont le cache est appliqué.

**Type**

Booléen ; lecture seule.

**Attribut AVLayer height**

`app.project.item(index).layer(index).height`

**Description**

Hauteur du calque, en pixels.

**Type**

Virgule flottante ; lecture seule.

**Attribut AVLayer isNameFromSource**

`app.project.item(index).layer(index).isNameFromSource`

**Description**

True si le calque ne possède pas de nom explicitement défini, mais contient une source nommée. Dans ce cas, *layer.name* possède la même valeur que *layer.source.name*.

False si le calque possède un nom explicitement défini ou si le calque ne possède pas de source.

**Type**

Booléen ; lecture seule.

**Attribut AVLayer isTrackMatte**

`app.project.item(index).layer(index).isTrackMatte`

**Description**

True si le calque est utilisé en tant que cache par approche pour le calque inférieur.

**Type**

Booléen ; lecture seule.

**Attribut AVLayer motionBlur**

`app.project.item(index).layer(index).motionBlur`

**Description**

True si le flou directionnel est activé pour le calque.

**Type**

Booléen ; lecture/écriture.

**Attribut AVLayer preserveTransparency**

`app.project.item(index).layer(index).preserveTransparency`

**Description**

True si l'option Préserver la transparence est activée pour le calque.

**Type**

Booléen ; lecture/écriture.

**Attribut AVLayer quality**

```
app.project.item(index).layer(index).quality
```

**Description**

Niveau de qualité selon lequel le calque doit s'afficher.

**Type**

Valeur énumérée LayerQuality ; lecture/écriture. L'une des suivantes :

```
LayerQuality.BEST
LayerQuality.DRAFT
LayerQuality.WIREFRAME
```

**Méthode AVLayer replaceSource()**

```
app.project.item(index).layer(index).replaceSource (newSource, fixExpressions)
```

**Description**

Remplace la source pour ce calque.

**Paramètres**

newSource	Nouvel objet AVItem source.
fixExpressions	True pour adapter les expressions à la nouvelle source, false dans le cas contraire. Notez que cette fonctionnalité peut solliciter de nombreuses ressources ; utilisez-la uniquement en fin d'opération en cas de remplacement d'une grande quantité de métrage. Voir aussi « Méthode Project autoFixExpressions() », page 111.

**Renvoi**

Aucun.

**Attribut AVLayer source**

```
app.project.item(index).layer(index).source
```

**Description**

AVItem source pour ce calque. La valeur est nulle dans un calque de texte. Utilisez AVLayer.replaceSource() pour modifier la valeur.

**Type**

Objet AVItem ; lecture seule.

**Méthode AVLayer sourceRectAtTime()**

```
app.project.item(index).layer(index).sourceRectAtTime(timeT, extents)
```

**Description**

Extrait les limites du rectangle du calque à l'index d'instant spécifié, corrigées selon le contenu du calque de texte ou de forme. Utilisez cette méthode pour écrire du texte correctement aligné sur la ligne de base, par exemple.

**Paramètres**

<code>timeT</code>	Index d'instant, en secondes. Valeur à virgule flottante.
<code>extents</code>	True pour inclure les extensions, false dans le cas contraire. Les extensions s'appliquent aux calques de forme et servent à augmenter la taille des limites du calque si nécessaire.

**Renvoi**

Un objet JavaScript avec quatre attributs, [top, left, width, height].

**Attribut AVLayer threeDLayer**

```
app.project.item(index).layer(index).threeDLayer
```

**Description**

True s'il s'agit d'un calque 3D.

**Type**

Booléen ; lecture/écriture.

**Attribut AVLayer threeDPerChar**

```
app.project.item(index).layer(index).threeDPerChar
```

**Description**

True si l'option d'activation de la projection 3D caractère par caractère est définie pour ce calque, ce qui permet d'animer ses caractères indépendamment du plan du calque de texte. S'applique uniquement aux calques de texte.

**Type**

Booléen ; lecture/écriture.

**Attribut AVLayer timeRemapEnabled**

```
app.project.item(index).layer(index).timeRemapEnabled
```

**Description**

True si le remappage temporel est activé pour ce calque.

**Type**

Booléen ; lecture/écriture.

**Attribut AVLayer trackMatteType**

`app.project.item(index).layer(index).trackMatteType`

**Description**

Si le calque comporte un cache par approche, spécifie la manière dont ce dernier sera appliqué.

**Type**

Valeur énumérée `TrackMatteType` ; lecture/écriture. L'une des suivantes :

`TrackMatteType.ALPHA`

`TrackMatteType.ALPHA_INVERTED`

`TrackMatteType.LUMA`

`TrackMatteType.LUMA_INVERTED`

`TrackMatteType.NO_TRACK_MATTE`

**Attribut AVLayer width**

`app.project.item(index).layer(index).width`

**Description**

Largeur du calque, en pixels.

**Type**

Virgule flottante ; lecture seule.

## Objet CameraLayer

`app.project.item(index).layer(index)`

### Description

L'objet CameraLayer représente un calque de caméra dans une composition. Vous pouvez le créer en utilisant la méthode `addCamera` de l'objet `LayerCollection` ; voir la section « Méthode `LayerCollection.addCamera()` », page 93. Pour accéder à cet objet dans la collection de calques d'un élément, vous pouvez utiliser son numéro d'index ou encore une chaîne de nom.

- CameraLayer est une sous-classe de Layer. Toutes les méthodes et tous les attributs de Layer sont disponibles lorsque vous utilisez CameraLayer. Voir la section « Objet Layer », page 83.

### Propriétés AE

CameraLayer ne définit aucun autre attribut, mais possède des propriétés AE différentes de celles des autres types de calque. Il dispose des propriétés et des groupes de propriétés suivants :

Marker

Transform

Point of Interest

Position

Scale

Orientation

X Rotation

Y Rotation

Rotation

Opacity

Camera Options

Zoom

Depth of Field

Focus Distance

Blur Level

## Objet Collection

De la même façon qu'un tableau, une collection regroupe un jeu d'objets ou de valeurs en tant que groupe logique, et fournit un accès à ces objets ou valeurs via un index. La plupart des objets collection sont toutefois en lecture seule. Vous ne leur attribuez pas les objets vous-même. Leur contenu est automatiquement mis à jour au fil des créations ou suppressions d'objets.

La numérotation d'index d'une collection commence par 1, et non par 0.

### Objets

Objet	Référence	Description
ItemCollection	« Objet ItemCollection », page 79	Tous les éléments (fichiers importés, dossiers, solides, etc.) se trouvant dans le panneau Projet.
LayerCollection	« Objet LayerCollection », page 92	Tous les calques d'une composition.
OMCollection	« Objet OMCollection », page 105	Tous les éléments de modules de sortie du projet.
RQItemCollection	« Objet RenderQueueItem », page 156	Tous les éléments de file d'attente de rendu du projet.

### Attributs

length	Nombre d'objets contenus dans la collection.
--------	--

### Méthodes

[ ]	Extrait un objet de la collection par le biais de son numéro d'index. Le premier objet se situe à l'index 1.
-----	--

## Objet Comptem

```
app.project.item(index)
app.project.items[index]
```

### Description

L'objet `Comptem` représente une composition et vous permet de la manipuler et d'obtenir des informations sur elle. Accédez aux objets d'un projet en utilisant leur numéro d'index dans la collection d'éléments `Item`.

- `Comptem` est une sous-classe d'`AVItem`, lui-même sous-classe d'`Item`. Toutes les méthodes et tous les attributs d'`AVItem` et d'`Item`, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez `Comptem`. Voir les sections « `Objet AVItem` », page 32 et « `Objet Item` », page 76.

### Exemple

Etant donné que le premier élément du projet est un `Comptem`, le code suivant affiche deux alertes. La première affiche le nombre de calques contenus dans le `Comptem`, et la deuxième le nom du dernier calque du `Comptem`.

```
var firstComp = app.project.item(1);
alert("number of layers is " + firstComp.numLayers);
alert("name of last layer is " + firstComp.layer(firstComp.numLayers).name);
```

### Attributs

Attribut	Référence	Description
<code>frameDuration</code>	« Attribut <code>Comptem frameDuration</code> », page 55	Durée d'une seule image.
<code>workAreaStart</code>	« Attribut <code>Comptem workAreaStart</code> », page 59	Heure de début de la zone de travail.
<code>workAreaDuration</code>	« Attribut <code>Comptem workAreaDuration</code> », page 59	Durée de la zone de travail.
<code>numLayers</code>	« Attribut <code>Comptem numLayers</code> », page 56	Nombre de calques contenus dans la composition.
<code>hideShyLayers</code>	« Attribut <code>Comptem hideShyLayers</code> », page 55	Lorsque cet attribut est <code>true</code> , les calques discrets s'affichent dans le panneau Montage.
<code>motionBlur</code>	« Attribut <code>Comptem motionBlur</code> », page 56	Lorsque cet attribut est <code>true</code> , le flou directionnel est activé pour cette composition.
<code>draft3d</code>	« Attribut <code>Comptem draft3d</code> », page 54	Lorsque cet attribut est <code>true</code> , le mode Ebauche 3D est activé pour le panneau Composition.
<code>frameBlending</code>	« Attribut <code>Comptem frameBlending</code> », page 54	Lorsque cet attribut est <code>true</code> , le filtrage temporel est activé pour cette composition.
<code>preserveNestedFrameRate</code>	« Attribut <code>Comptem preserveNestedFrameRate</code> », page 56	Lorsque cet attribut est <code>true</code> , la cadence d'image des compositions imbriquées est préservée.
<code>preserveNestedResolution</code>	« Attribut <code>Comptem preserveNestedResolution</code> », page 57	Lorsque cet attribut est <code>true</code> , la résolution des compositions imbriquées est préservée.
<code>bgColor</code>	« Attribut <code>Comptem bgColor</code> », page 53	Couleur de fond de la composition.
<code>activeCamera</code>	« Attribut <code>Comptem activeCamera</code> », page 53	Calque de caméra actif.

Attribut	Référence	Description
displayStartTime	« Attribut Comptem displayStartTime », page 54	Modifie l'affichage de l'heure de début dans le panneau Montage.
resolutionFactor	« Attribut Comptem resolutionFactor », page 57	Facteur par lequel la résolution x et y du panneau Composition est sous-échantillonnée.
shutterAngle	« Attribut Comptem shutterAngle », page 58	Angle obturateur de la caméra.
shutterPhase	« Attribut Comptem shutterPhase », page 58	Phase d'obturateur de la caméra.
layers	« Attribut Comptem layers », page 56 « Objet LayerCollection », page 92	Calques de la composition.
selectedLayers	« Attribut Comptem selectedLayers », page 58	Calques sélectionnés de la composition.
selectedProperties	« Attribut Comptem selectedProperties », page 58	Propriétés sélectionnées de la composition.
renderer	« Attribut Comptem renderer », page 57	Module externe de rendu à utiliser pour le rendu de cette composition.
renderers	« Attribut Comptem renderers », page 57	Ensemble des modules externes de rendu disponibles.

#### Méthodes

Méthode	Référence	Description
duplicate()	« Méthode Comptem duplicate() », page 54	Crée et renvoie un doublon de cette composition.
layer()	« Méthode Comptem layer() », page 55	Obtient un calque à partir de cette composition.

#### Attribut Comptem activeCamera

`app.project.item(index).activeCamera`

##### Description

Il s'agit de la caméra active, c'est-à-dire du calque de caméra activé se trouvant au premier plan. La valeur renvoyée est nulle si la composition ne contient aucun calque de caméra activé.

##### Type

Objet CameraLayer ; lecture seule.

#### Attribut Comptem bgColor

`app.project.item(index).bgColor`

##### Description

Couleur de fond de la composition. Les trois valeurs du tableau définissent les composantes rouge, vert et bleu de la couleur.

##### Type

Tableau contenant trois valeurs à virgule flottante, [R, V, B], comprises dans la plage [0.0..1.0] ; lecture/écriture.

**Attribut ComItem displayStartTime**

`app.project.item(index).displayStartTime`

**Description**

Heure définie en tant que début de la composition, en secondes. Elle est équivalente à la valeur Code temporel initial ou Première image définie dans la boîte de dialogue Paramètres de composition.

**Type**

Valeur à virgule flottante comprise dans la plage [0.0...86339.0] (25 heures moins 1 seconde) ; lecture/écriture.

**Attribut ComItem draft3d**

`app.project.item(index).draft3d`

**Description**

Lorsque cet attribut est true, le mode Ebauche 3D est activé pour le panneau Composition. Sa valeur correspond à celle du bouton Ebauche 3D du panneau Composition.

**Type**

Booléen ; lecture/écriture.

**Méthode ComItem duplicate()**

`app.project.item(index).duplicate()`

**Description**

Crée et renvoie un doublon de cette composition, contenant les mêmes calques que l'original.

**Paramètres**

Aucun.

**Renvoi**

Objet ComItem.

**Attribut ComItem frameBlending**

`app.project.item(index).frameBlending`

**Description**

Lorsque cet attribut est true, l'interpolation est activée pour cette composition. Sa valeur correspond à celle du bouton d'interpolation dans le panneau Composition.

**Type**

Booléen ; si true, l'interpolation d'images est activée ; lecture/écriture.

**Attribut Comptem frameDuration**

```
app.project.item(index).frameDuration
```

**Description**

Durée d'une image, exprimée en secondes. Il s'agit de l'inverse de la valeur `frameRate` (images par seconde).

**Type**

Virgule flottante ; lecture/écriture.

**Attribut Comptem hideShyLayers**

```
app.project.item(index).hideShyLayers
```

**Description**

Lorsque cet attribut est `true`, seuls les calques pour lesquels `shy` est défini sur `false` s'affichent dans le panneau Montage. Lorsqu'il est `false`, tous les calques sont visibles, y compris ceux pour lesquels la valeur `shy` est `true`. Correspond à la valeur du bouton Masquer calques discrets du panneau Composition.

**Type**

Booléen ; lecture/écriture.

**Méthode Comptem layer()**

```
app.project.item(index).layer(index)
app.project.item(index).layer(otherLayer, relIndex)
app.project.item(index).layer(name)
```

**Description**

Renvoie un objet `Layer`, pouvant être spécifié à l'aide d'un nom, d'une position d'index dans ce calque ou d'une position d'index relative à un autre calque.

**Paramètres**

<code>index</code>	Numéro d'index du calque souhaité dans cette composition. Nombre entier compris dans la plage [1... <i>numLayers</i> ], où <i>numLayers</i> correspond au nombre de calques de la composition.
--------------------	--

—ou—

<code>otherLayer</code>	Objet <code>Layer</code> dans cette composition. La valeur <code>relIndex</code> est ajoutée à la valeur d'index de ce calque afin de trouver la position du calque souhaité.
<code>relIndex</code>	Position du calque souhaité, relative à <code>otherLayer</code> . Nombre entier compris dans la plage [1- <code>otherLayer.index</code> ... <i>numLayers</i> - <code>otherLayer.index</code> ], où <i>numLayers</i> correspond au nombre de calques de la composition.  Cette valeur est ajoutée à la valeur <code>otherLayer</code> afin d'obtenir l'index absolu du calque à renvoyer.

—ou—

<code>name</code>	Chaîne contenant le nom du calque souhaité.
-------------------	---

**Renvoi**

Objet Layer.

**Attribut Comptem layers**

`app.project.item(index).layers`

**Description**

Objet LayerCollection qui contient tous les objets Layer des calques de cette composition. Voir la section « Objet LayerCollection », page 92.

**Type**

Objet LayerCollection ; lecture seule.

**Attribut Comptem motionBlur**

`app.project.item(index).motionBlur`

**Description**

Lorsque cet attribut est true, le flou directionnel est activé pour la composition. Sa valeur correspond à celle du bouton Flou directionnel du panneau Composition.

**Type**

Booléen ; lecture/écriture.

**Attribut Comptem numLayers**

`app.project.item(index).numLayers`

**Description**

Nombre de calques contenus dans la composition.

**Type**

Entier ; lecture seule.

**Attribut Comptem preserveNestedFrameRate**

`app.project.item(index).preserveNestedFrameRate`

**Description**

Lorsque cet attribut est true, la cadence d'image des compositions imbriquées est préservée dans la composition en cours. Correspond à la valeur de l'option « Préserver la cadence (compositions imbriquées ou en file d'attente de rendu) » de l'onglet Avancé de la boîte de dialogue Paramètres de composition.

**Type**

Booléen ; lecture/écriture.

**Attribut Comptem preserveNestedResolution**

```
app.project.item(index).preserveNestedResolution
```

**Description**

Lorsque cet attribut est true, la résolution des compositions imbriquées est préservée dans la composition en cours. Correspond à la valeur de l'option « Préserver la résolution (compositions imbriquées) » de l'onglet Avancé de la boîte de dialogue Paramètres de composition.

**Type**

Booléen ; lecture/écriture.

**Attribut Comptem renderer**

```
app.project.item(index).renderer
```

**Description**

Module externe de rendu à utiliser pour le rendu de cette composition, comme indiqué dans l'onglet Avancé de la boîte de dialogue Paramètres de composition. Les valeurs autorisées sont les membres de *compItem.renderers*.

**Type**

Chaîne ; lecture/écriture.

**Attribut Comptem renderers**

```
app.project.item(index).renderers
```

**Description**

Modules externes de rendu disponibles. Les chaînes des membres correspondent aux modules installés, comme indiqué dans l'onglet Avancé de la boîte de dialogue Paramètres de composition.

**Type**

Tableau de chaînes ; lecture seule.

**Attribut Comptem resolutionFactor**

```
app.project.item(index).resolutionFactor
```

**Description**

Facteurs de résolution de sous-échantillonnage de *x* et *y* pour le rendu de la composition.

Les deux valeurs du tableau indiquent le nombre de pixels à ignorer lors de l'échantillonnage. Le premier numéro détermine l'échantillonnage horizontal, le second l'échantillonnage vertical. La résolution intégrale est [1,1], la demi-résolution [2,2] et la résolution d'un quart [4,4]. La valeur par défaut est [1,1].

**Type**

Tableau de deux entiers compris dans la plage de valeurs [1..99] ; lecture/écriture.

**Attribut Comptem selectedLayers**

`app.project.item(index).selectedLayers`

**Description**

Tous les calques sélectionnés de cette composition. Il s'agit d'un tableau qui a zéro (0) pour base (le premier objet se situe à l'index 0).

**Type**

Tableau d'objets Layer ; lecture seule.

**Attribut Comptem selectedProperties**

`app.project.item(index).selectedProperties`

**Description**

Toutes les propriétés sélectionnées (objets Property et PropertyGroup) dans cette composition. La première propriété se situe à l'index 0.

**Type**

Tableau d'objets Property et PropertyGroup ; lecture seule.

**Attribut Comptem shutterAngle**

`app.project.item(index).shutterAngle`

**Description**

Réglage de l'angle obturateur pour la composition. Ce réglage correspond au paramètre Angle obturateur de l'onglet Avancé de la boîte de dialogue Paramètres de composition.

**Type**

Nombre entier compris dans la plage [0...720] ; lecture/écriture.

**Attribut Comptem shutterPhase**

`app.project.item(index).shutterPhase`

**Description**

Réglage de la phase d'obturateur pour la composition. Ce réglage correspond au paramètre Phase d'obturateur de l'onglet Avancé de la boîte de dialogue Paramètres de composition.

**Type**

Nombre entier compris dans la plage [-360...360] ; lecture/écriture.

**Attribut Comptem workAreaDuration**

`app.project.item(index).workAreaDuration`

**Description**

Durée de la zone de travail, en secondes. Il s'agit de la différence entre l'instant du point de début et l'instant du point de fin de la zone de travail Composition.

**Type**

Virgule flottante ; lecture/écriture.

**Attribut Comptem workAreaStart**

`app.project.item(index).workAreaStart`

**Description**

Moment, en secondes, où la zone de travail Composition commence.

**Type**

Virgule flottante ; lecture/écriture.

## Objet FileSource

`app.project.item(index).mainSource`

`app.project.item(index).proxySource`

### Description

L'objet FileSource décrit le métrage issu d'un fichier.

- FileSource est une sous-classe de FootageSource. Toutes les méthodes et tous les attributs de FootageSource, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez FileSource. Voir la section « Objet FootageSource », page 67.

### Attributs

Attribut	Référence	Description
file	« Attribut FileSource file », page 60	Fichier qui définit cette ressource.
missingFootagePath	« Attribut FileSource missingFootagePath », page 60	Fichier contenant le métrage absent de cette ressource.

### Méthodes

Méthode	Référence	Description
reload()	« Méthode FileSource reload() », page 61	Recharge la ressource du fichier, s'il s'agit d'un mainSource d'un FootageItem.

### Attribut FileSource file

`app.project.item(index).mainSource.file`

`app.project.item(index).proxySource.file`

### Description

Objet ExtendScript File pour le fichier définissant cette ressource. Pour modifier la valeur :

- Si ce FileSource est un proxySource d'un AVItem, appelez `setProxy()` ou `setProxyWithSequence()`.
- Si ce FileSource est un mainSource d'un FootageItem, appelez `replace()` ou `replaceWithSequence()`.

### Type

Objet File ; lecture seule.

### Attribut FileSource missingFootagePath

`app.project.item(index).mainSource.file.missingFootagePath`

`app.project.item(index).proxySource.file.missingFootagePath`

### Description

Chemin et nom de fichier du métrage absent de cette ressource. Voir aussi la section « Attribut AVItem footageMissing », page 33.

### Type

Chaîne ; lecture seule.

**Méthode FileSource reload()**

```
app.project.item(index).mainSource.file.mainSource.reload()
```

**Description**

Recharge la ressource du fichier. Cette méthode peut uniquement être appelée sur un `mainSource`, et non sur un `proxySource`.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

## Objet FolderItem

app.project.FolderItem

### Description

L'objet FolderItem correspond à un dossier de votre panneau Projet. Il peut contenir différents types d'éléments (métrage, compositions, solides), ainsi que d'autres dossiers.

### Exemple

Etant donné que le deuxième élément du projet est un FolderItem, le code suivant génère une alerte pour chaque élément de niveau supérieur dans le dossier, en affichant son nom.

```
var secondItem = app.project.item(2);
if ( !(secondItem instanceof FolderItem) ) {
  alert("problem: second item is not a folder");
} else {
  for ( i = 1; i <= secondItem.numItems; i++ ) {
    alert("item number " + i + " within the folder is named "
      + secondItem.item(i).name);
  }
}
```

### Attributs

Attribut	Référence	Description
items	« Attribut FolderItem items », page 63	Contenu de ce dossier.
numItems	« Attribut FolderItem numItems », page 63	Nombre d'éléments contenus dans ce dossier.

### Méthodes

Méthode	Référence	Description
item()	« Méthode FolderItem item() », page 62	Récupère un élément du dossier.

### Méthode FolderItem item()

app.project.item(*index*).item

### Description

Renvoie l'élément de niveau supérieur de ce dossier à la position d'index spécifiée. Notez que la notion de « niveau supérieur » fait référence aux niveaux à l'intérieur du dossier, mais pas nécessairement dans le projet.

### Paramètres

index	Nombre entier, numéro d'index de l'élément à récupérer. Le premier élément se situe à l'index 1.
-------	--

### Renvoi

Objet Item.

**Attribut FolderItem items**

```
app.project.item(index).items
```

**Description**

Objet ItemCollection qui contient un objet Item représentant le contenu de niveau supérieur de ce dossier.

Contrairement à ItemCollection dans l'objet Project, cette collection contient uniquement les éléments de niveau supérieur du dossier. Le niveau supérieur dans le dossier ne correspond pas au niveau supérieur dans le projet. Seuls les éléments situés au niveau supérieur du dossier racine se retrouvent également au niveau supérieur dans le projet.

**Type**

Objet ItemCollection ; lecture seule.

**Attribut FolderItem numItems**

```
app.project.item(index).numItems
```

**Description**

Nombre d'éléments contenus dans la collection d'éléments (*folderItem.items.length*).

Si le dossier contient un autre dossier, seul le FolderItem pour ce dossier est pris en compte. Les sous-éléments qu'il contient sont ignorés.

**Type**

Entier ; lecture seule.

## Objet FootageItem

```
app.project.item(index)
app.project.items[index]
```

### Description

L'objet FootageItem représente un élément de métrage importé dans un projet, qui apparaît dans le panneau Projet. Il est possible d'accéder à ces objets dans la collection d'éléments `Item` d'un projet en utilisant leur numéro d'index.

- FootageItem est une sous-classe d'AVItem, lui-même sous-classe d'Item. Toutes les méthodes et tous les attributs d'AVItem et d'Item, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez FootageItem. Voir les sections « Objet AVItem », page 32 et « Objet Item », page 76.

### Attributs

Attribut	Référence	Description
<code>file</code>	« Attribut FootageItem file », page 64	Fichier source du métrage.
<code>mainSource</code>	« Attribut FootageItem mainSource », page 65	Tous les paramètres liés à l'élément de métrage.

### Méthodes

Méthode	Référence	Description
<code>replace()</code>	« Méthode FootageItem replace() », page 65	Remplace un fichier de métrage par un autre fichier de métrage.
<code>replaceWithPlaceholder()</code>	« Méthode FootageItem replaceWithPlaceholder() », page 65	Remplace un fichier de métrage par un objet pseudo-élément.
<code>replaceWithSequence()</code>	« Méthode FootageItem replaceWithSequence() », page 66	Remplace un fichier de métrage par une séquence d'images.
<code>replaceWithSolid()</code>	« Méthode FootageItem replaceWithSolid() », page 66	Remplace un fichier de métrage par un solide.

### Attribut FootageItem file

```
app.project.item(index).file
```

### Description

Objet ExtendScript File pour le fichier source du métrage.

Si le `mainSource` du FootageItem est un FileSource, ce fichier est le même que `FootageItem.mainSource.file`. Sinon, la valeur est nulle.

### Type

Objet File ; lecture seule.

**Attribut FootageItem mainSource**

`app.project.item(index).mainSource`

**Description**

Source du métrage, objet qui contient tous les paramètres liés à cet élément de métrage, y compris ceux habituellement accessibles par le biais de la boîte de dialogue Interpréter un métrage. L'attribut est en lecture seule. Pour modifier sa valeur, appelez l'une des méthodes FootageItem « replace ».

Reportez-vous à la section « Objet FootageSource », page 67 et aux trois types d'objets différents :

- « Objet SolidSource », page 170
- « Objet FileSource », page 60
- « Objet PlaceholderSource », page 109

S'il s'agit d'un objet FileSource et que la valeur de `footageMissing` est true, le chemin d'accès au fichier de métrage manquant se trouve dans l'attribut `FileSource.missingFootagePath`. Voir les sections « Attribut AVItem footageMissing », page 33 et « Attribut FileSource missingFootagePath », page 60.

**Type**

Objet FootageSource ; lecture seule.

**Méthode FootageItem replace()**

`app.project.item(index).replace(file)`

**Description**

Remplace la source de ce FootageItem par le fichier spécifié. En plus de charger le fichier, la méthode crée un nouvel objet FileSource pour le fichier et définit `mainSource` sur cet objet. Dans le nouvel objet source, elle définit les attributs `name`, `width`, `height`, `frameDuration` et `duration` (voir la section « Objet AVItem », page 32) en fonction du contenu du fichier.

La méthode préserve les paramètres d'interprétation de l'objet `mainSource` précédent. Si le fichier spécifié comporte une couche alpha non identifiée, la méthode évalue l'interprétation alpha.

**Paramètres**

file	Objet ExtendScript File pour le fichier à utiliser en tant que source principale du métrage.
------	--

**Méthode FootageItem replaceWithPlaceholder()**

`app.project.item(index).replaceWithPlaceholder(name, width, height, frameRate, duration)`

**Description**

Remplace la source de ce FootageItem par le pseudo-élément spécifié. Crée un nouvel objet PlaceholderSource, définit ses valeurs à partir des paramètres et définit `mainSource` sur cet objet.

**Paramètres**

name	Chaîne contenant le nom du pseudo-élément.
width	Largeur du pseudo-élément, en pixels. Nombre entier compris dans la plage [4..30000].
height	Hauteur du pseudo-élément, en pixels. Nombre entier compris dans la plage [4..30000].

<code>frameRate</code>	Cadence du pseudo-élément. Valeur à virgule flottante comprise dans la plage [1.0..99.0].
<code>duration</code>	Durée du pseudo-élément, en secondes. Valeur à virgule flottante comprise dans la plage [0.0..10800.0].

### Méthode FootageItem `replaceWithSequence()`

`app.project.item(index).replaceWithSequence(file, forceAlphabetical)`

#### Description

Remplace la source de ce FootageItem par la séquence d'images spécifiée. En plus de charger le fichier, la méthode crée un nouvel objet FileSource pour le fichier et définit `mainSource` sur cet objet. Dans le nouvel objet source, elle définit les attributs `name`, `width`, `height`, `frameDuration` et `duration` (voir la section « Objet AVItem », page 32) en fonction du contenu du fichier.

La méthode préserve les paramètres d'interprétation de l'objet `mainSource` précédent. Si le fichier spécifié comporte une couche alpha non identifiée, la méthode évalue l'interprétation alpha.

#### Paramètres

<code>file</code>	Objet ExtendScript File pour le premier fichier de la séquence à utiliser en tant que source principale du métrage.
<code>forceAlphabetical</code>	Lorsque ce paramètre est true, l'option Ordre alphabétique doit être utilisée.

### Méthode FootageItem `replaceWithSolid()`

`app.project.item(index).replaceWithSolid(color, name, width, height, pixelAspect)`

#### Description

Remplace la source de ce FootageItem par le solide spécifié. Crée un nouvel objet SolidSource, définit ses valeurs à partir des paramètres et définit `mainSource` sur cet objet.

#### Paramètres

<code>color</code>	Couleur du solide, tableau de trois valeurs à virgule flottante, [R, V, B], comprises dans la plage [0.0..1.0].
<code>name</code>	Chaîne contenant le nom du solide.
<code>width</code>	Largeur du solide, en pixels. Nombre entier compris dans la plage [4..30000].
<code>height</code>	Hauteur du solide, en pixels. Nombre entier compris dans la plage [4..30000].
<code>pixelAspect</code>	Format des pixels du solide. Valeur à virgule flottante comprise dans la plage [0.01..100.0].

## Objet FootageSource

`app.project.item(index).mainSource`

`app.project.item(index).proxySource`

### Description

L'objet FootageSource contient des informations décrivant la source d'un métrage. Il est utilisé en tant que `mainSource` d'un FootageItem ou en tant que `proxySource` d'un CompItem ou d'un FootageItem. Voir les sections « Objets FootageItem », page 64 et « Objets CompItem », page 52.

- FootageSource correspond à la classe de base de SolidSource. Ainsi, les attributs et les méthodes FootageSource sont disponibles lorsque vous utilisez des objets SolidSource. Voir la section « Objets SolidSource », page 170.

### Attributs

Attribut	Référence	Description
<code>hasAlpha</code>	« Attribut FootageSource <code>hasAlpha</code> », page 70	Lorsque cet attribut est true, une séquence ou une doublure de métrage inclut une couche alpha.
<code>alphaMode</code>	« Attribut FootageSource <code>alphaMode</code> », page 68	Mode d'une couche alpha.
<code>premulColor</code>	« Attribut FootageSource <code>premulColor</code> », page 71	Couleur à prémultiplier.
<code>invertAlpha</code>	« Attribut FootageSource <code>invertAlpha</code> », page 70	Lorsque cet attribut est true, une couche alpha d'une séquence de métrage ou d'une doublure doit être inversée.
<code>isStill</code>	« Attribut FootageSource <code>isStill</code> », page 70	Lorsque cet attribut est true, le métrage est une image fixe.
<code>fieldSeparationType</code>	« Attribut FootageSource <code>fieldSeparationType</code> », page 69	Type de séparation de trame.
<code>highQualityFieldSeparation</code>	« Attribut FootageSource <code>highQualityFieldSeparation</code> », page 70	Définit la façon dont les trames doivent être séparées dans un métrage non fixe.
<code>removePullDown</code>	« Attribut FootageSource <code>removePullDown</code> », page 72	Type de conversion du métrage.
<code>loop</code>	« Attribut FootageSource <code>loop</code> », page 71	Nombre de boucles d'une séquence d'images.
<code>nativeFrameRate</code>	« Attribut FootageSource <code>nativeFrameRate</code> », page 71	Cadence native du métrage.
<code>displayFrameRate</code>	« Attribut FootageSource <code>displayFrameRate</code> », page 68	Cadence effective telle qu'affichée et rendue par After Effects dans les compositions.
<code>conformFrameRate</code>	« Attribut FootageSource <code>conformFrameRate</code> », page 68	Cadence à laquelle le métrage doit se conformer.

### Méthodes

Méthode	Référence	Description
<code>guessAlphaMode()</code>	« Méthode FootageSource <code>guessAlphaMode()</code> », page 69	Evalue le paramètre <code>alphaMode</code> .
<code>guessPullDown()</code>	« Méthode FootageSource <code>guessPullDown()</code> », page 69	Evalue le paramètre <code>pullDownType</code> .

**Attribut FootageSource alphaMode**

`app.project.item(index).mainSource.alphaMode`  
`app.project.item(index).proxySource.alphaMode`

**Description**

L'attribut `alphaMode` de la `footageSource` définit la façon dont les informations alpha contenues dans le métrage doivent être interprétées. Si `hasAlpha` est `false`, cet attribut est sans effet.

**Type**

Valeur énumérée `AlphaMode` ; lecture/écriture. L'une des suivantes :

`AlphaMode.IGNORE`  
`AlphaMode.STRAIGHT`  
`AlphaMode.PREMULTIPLIED`

**Attribut FootageSource conformFrameRate**

`app.project.item(index).mainSource.conformFrameRate`  
`app.project.item(index).proxySource.conformFrameRate`

**Description**

Cadence d'image à utiliser à la place de la valeur `nativeFrameRate`. Si cet attribut est défini sur 0, la valeur `nativeFrameRate` est utilisée à la place.

Vous ne devez pas définir cette valeur si `FootageSource.isStill` est `true`. Vous ne devez pas définir cette valeur sur 0 si `removePullDown` n'est pas défini sur `PullDownPhase.OFF`. Si la valeur est 0 lorsque vous définissez `removePullDown` sur une valeur différente de `PullDownPhase.OFF`, cette valeur est automatiquement définie sur la valeur de `nativeFrameRate`.

**Type**

Valeur à virgule flottante comprise dans la plage [0.0.. 99.0] ; lecture/écriture.

**Attribut FootageSource displayFrameRate**

`app.project.item(index).mainSource.displayFrameRate`  
`app.project.item(index).proxySource.displayFrameRate`

**Description**

Cadence effective telle qu'affichée et rendue par `After Effects` dans les compositions.

Si `removePullDown` est égal à `PullDownPhase.OFF`, cela équivaut à la valeur `conformFrameRate` (si elle est différente de zéro) ou à la valeur `nativeFrameRate` (si `conformFrameRate` est égal à 0). Si `removePullDown` est différent de `PullDownPhase.OFF`, cela équivaut à la valeur `conformFrameRate * 0,8`, la cadence réelle après suppression d'une image sur cinq.

**Type**

Valeur à virgule flottante comprise dans la plage [0.0.. 99.0] ; lecture seule.

**Attribut FootageSource fieldSeparationType**

app.project.item(*index*).mainSource.fieldSeparationType  
 app.project.item(*index*).proxySource.fieldSeparationType

**Description**

Définit la façon dont les trames doivent être séparées dans un métrage non fixe.

Vous ne devez pas définir cet attribut si isStill est true. Vous ne devez pas définir cette valeur sur FieldSeparationType.OFF si removePulldown est différent de PulldownPhase.OFF.

**Type**

Valeur énumérée FieldSeparationType ; lecture/écriture. L'une des suivantes :

FieldSeparationType.OFF  
 FieldSeparationType.UPPER\_FIELD\_FIRST  
 FieldSeparationType.LOWER\_FIELD\_FIRST

**Méthode FootageSource guessAlphaMode()**

app.project.item(*index*).mainSource.guessAlphaMode()  
 app.project.item(*index*).proxySource.guessAlphaMode()

**Description**

Définit alphaMode, premulColor et invertAlpha sur les évaluations les plus précises pour cette source de métrage. Si hasAlpha est false, aucun changement n'est effectué.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Méthode FootageSource guessPulldown()**

app.project.item(*index*).mainSource.guessPulldown(*method*)  
 app.project.item(*index*).proxySource.guessPulldown(*method*)

**Description**

Définit fieldSeparationType et removePulldown sur les évaluations les plus précises pour cette source de métrage. Si isStill est true, aucun changement n'est effectué.

**Paramètres**

method	Méthode à utiliser pour l'estimation. Valeur énumérée PulldownMethod, l'une des suivantes : PulldownMethod.PULLDOWN_3_2 PulldownMethod.ADVANCE_24P
--------	--

**Renvoi**

Aucun.

**Attribut FootageSource hasAlpha**

`app.project.item(index).mainSource.hasAlpha`  
`app.project.item(index).proxySource.hasAlpha`

**Description**

Lorsque cet attribut est défini sur `true`, le métrage possède un composant alpha. Dans ce cas, les attributs `alphaMode`, `invertAlpha` et `premulColor` ont des valeurs valides. Lorsqu'il est défini sur `false`, ces attributs n'ont pas d'importance pour le métrage.

**Type**

Booléen ; lecture seule.

**Attribut FootageSource highQualityFieldSeparation**

`app.project.item(index).mainSource.highQualityFieldSeparation`  
`app.project.item(index).proxySource.highQualityFieldSeparation`

**Description**

Lorsque cet attribut est défini sur `true`, After Effects utilise des algorithmes particuliers pour déterminer la façon d'effectuer une séparation de trame haute qualité.

Vous ne devez pas définir cet attribut si `isStill` a la valeur `true` ou si `fieldSeparationType` est `FieldSeparationType.OFF`.

**Type**

Booléen ; lecture/écriture.

**Attribut FootageSource invertAlpha**

`app.project.item(index).mainSource.invertAlpha`  
`app.project.item(index).proxySource.invertAlpha`

**Description**

Lorsque cet attribut est `true`, une couche alpha d'une séquence de métrage ou d'une doublure doit être inversée.

Cet attribut n'est valide qu'en présence d'une couche alpha. Si `hasAlpha` a la valeur `false` ou que `alphaMode` est `AlphaMode.IGNORE`, cet attribut est ignoré.

**Type**

Booléen ; lecture/écriture.

**Attribut FootageSource isStill**

`app.project.item(index).mainSource.isStill`  
`app.project.item(index).proxySource.isStill`

**Description**

Lorsque cet attribut est défini sur `true`, le métrage est fixe. Lorsqu'il est défini sur `false`, le métrage possède un composant temporel.

Les fichiers JPEG, les solides et les pseudo-éléments d'une durée de 0 sont des métrages fixes. Les fichiers de films, les fichiers son, les séquences et les pseudo-éléments d'une durée différente de zéro sont des métrages non fixes.

**Type**

Booléen ; lecture seule.

**Attribut FootageSource loop**

`app.project.item(index).mainSource.loop`  
`app.project.item(index).proxySource.loop`

**Description**

Nombre de lectures consécutives du métrage lors de son utilisation dans une composition.

Vous ne devez pas définir cet attribut si `isStill` est true.

**Type**

Nombre entier compris dans la plage [1.. 9999] ; la valeur par défaut est 1 ; lecture/écriture.

**Attribut FootageSource nativeFrameRate**

`app.project.item(index).mainSource.nativeFrameRate`  
`app.project.item(index).proxySource.nativeFrameRate`

**Description**

Cadence native du métrage.

**Type**

Virgule flottante ; lecture/écriture.

**Attribut FootageSource premulColor**

`app.project.item(index).mainSource.premulColor`  
`app.project.item(index).proxySource.premulColor`

**Description**

Couleur à prémultiplier. Cet attribut n'est valide que si `alphaMode` est défini sur `alphaMode.PREMULTIPLIED`.

**Type**

Tableau de trois valeurs à virgule flottante, [R, G, B], comprises dans la plage [0.0..1.0] ; lecture/écriture.

**Attribut FootageSource removePulldown**

```
app.project.item(index).mainSource.removePulldown  
app.project.item(index).proxySource.removePulldown
```

**Description**

Manière dont les conversions doivent être supprimées lorsque la séparation de trame est utilisée.

Vous ne devez pas définir cet attribut si `isStill` est true. Vous ne devez pas définir cet attribut sur une valeur différente de `PulldownPhase.OFF` si `fieldSeparationType` est `FieldSeparationType.OFF`.

**Type**

Valeur énumérée `PulldownPhase` ; lecture/écriture. L'une des suivantes :

```
PulldownPhase.RemovePulldown.OFF  
PulldownPhase.RemovePulldown.WSSWW  
PulldownPhase.RemovePulldown.SSWWW  
PulldownPhase.RemovePulldown.SWWWS  
PulldownPhase.RemovePulldown.WWWSS  
PulldownPhase.RemovePulldown.WWSSW  
PulldownPhase.RemovePulldown.WSSWW_24P_ADVANCE  
PulldownPhase.RemovePulldown.SSWWW_24P_ADVANCE  
PulldownPhase.RemovePulldown.SWWWS_24P_ADVANCE  
PulldownPhase.RemovePulldown.WWWSS_24P_ADVANCE  
PulldownPhase.RemovePulldown.WWSSW_24P_ADVANCE
```

## Objet ImportOptions

```
new ImportOptions();
new ImportOptions(file);
```

### Description

L'objet ImportOptions incorpore les options utilisées pour importer un fichier à l'aide des méthodes Project.importFile. Voir la section « Méthode Project importFile() », page 113.

Le constructeur prend un paramètre facultatif, un objet ExtendScript File pour le fichier. S'il n'est pas fourni, vous devez définir explicitement la valeur de l'attribut file avant d'utiliser l'objet avec la méthode importFile. Par exemple :

```
new ImportOptions().file = new File("myfile.psd");
```

### Attributs

Attributs	Référence	Description
importAs	« Attribut ImportOptions importAs », page 74	Type de fichier à importer.
sequence	« Attribut ImportOptions sequence », page 75	Lorsque cet attribut est défini sur true, importe une séquence de fichiers plutôt qu'un seul fichier.
forceAlphabetical	« Attribut ImportOptions forceAlphabetical », page 74	Lorsque cet attribut est défini sur true, l'option Ordre alphabétique est définie.
file	« Attribut ImportOptions file », page 74	Fichier à importer ou premier fichier de la séquence à importer.

### Méthodes

Méthode	Référence	Description
canImportAs()	« Méthode ImportOptions canImportAs() », page 73	Limite l'entrée à un type de fichier particulier.

### Méthode ImportOptions canImportAs()

```
importOptions.canImportAs(type)
```

### Description

Indique si le fichier peut être importé en tant que source d'un type d'objet particulier. Si la méthode renvoie true, vous pouvez définir le type donné en tant que valeur de l'attribut importAs. Voir la section « Attribut ImportOptions importAs », page 74.

### Paramètres

type	Type de fichier pouvant être importé. Valeur énumérée ImportAsType ; l'une des suivantes : ImportAsType.COMP ImportAsType.FOOTAGE ImportAsType.COMP_CROPPED_LAYERS ImportAsType.PROJECT
------	---

### Renvoi

Booléen.

**Exemple**

```
var io = new ImportOptions(File("c:\\myFile.psd"));
if io.canImportAs(ImportAsType.COMP);
    io.importAs = ImportAsType.COMP;
```

**Attribut ImportOptions file**

*importOptions.file*

**Description**

Fichier à importer. Si un fichier est défini dans le constructeur, vous pouvez y accéder via cet attribut.

**Type**

Objet ExtendScript File ; lecture/écriture.

**Attribut ImportOptions forceAlphabetical**

*importOptions.forceAlphabetical*

**Description**

Lorsque cet attribut est défini sur true, produit le même effet que lorsque l'option Ordre alphabétique de la boîte de dialogue Fichier > Importer > Fichier est activée.

**Type**

Booléen ; lecture/écriture.

**Attribut ImportOptions importAs**

*importOptions.importAs*

**Description**

Type d'objet pour lequel le fichier importé sert de source. Avant de le définir, utilisez `canImportAs` pour vérifier qu'un fichier donné peut être importé en tant que source du type d'objet donné. Voir la section « Méthode ImportOptions canImportAs() », page 73.

**Type**

Valeur énumérée ImportAsType ; lecture/écriture. L'une des suivantes :

ImportAsType.COMP\_CROPPED\_LAYERS

ImportAsType.FOOTAGE

ImportAsType.COMP

ImportAsType.PROJECT

**Attribut ImportOptions sequence***importOptions.sequence***Description**

Lorsque cet attribut est défini sur true, une séquence est importée ; sinon, un fichier seul est importé.

**Type**

Booléen ; lecture/écriture.

## Objet Item

```
app.project.item(index)
app.project.items[index]
```

### Description

L'objet Item représente un élément pouvant apparaître dans le panneau Projet.

Le premier élément se situe à l'index 1.

- Item est la classe de base d'AVItem et de FolderItem, eux-mêmes classes de base de plusieurs autres types d'élément. Ainsi, les attributs et les méthodes Item sont disponibles lorsque vous utilisez ces types d'élément. Voir les sections « Objet AVItem », page 32 et « Objet FolderItem », page 62.

### Attributs

Attributs	Référence	Description
name	« Attribut Item name », page 77	Nom de l'objet, tel qu'affiché dans le panneau Projet.
comment	« Attribut Item comment », page 77	Chaîne descriptive.
id	« Attribut Item id », page 77	Identificateur unique pour cet élément.
parentFolder	« Attribut Item parentFolder », page 77	Dossier parent de cet élément.
selected	« Attribut Item selected », page 78	Lorsque cet attribut est défini sur true, cet élément est actuellement sélectionné.
typeName	« Attribut Item typeName », page 78	Type d'élément.

### Méthodes

Méthode	Référence	Description
remove()	« Méthode Item remove() », page 78	Supprime l'élément du projet.

### Exemple

L'exemple suivant obtient le deuxième élément du projet et vérifie qu'il s'agit d'un dossier. Il supprime ensuite du dossier tout élément de niveau supérieur n'étant pas actuellement sélectionné. Il vérifie par ailleurs, pour chaque élément du dossier, que le parent est correctement défini sur le dossier adéquat.

```
var myFolder = app.project.item(2);
if (myFolder.typeName != "Folder") {
    alert("error: second item is not a folder");
}
else {
    var numInFolder = myFolder.numItems;
    // Always run loops backwards when deleting things:
    for(i = numInFolder; i >= 1; i--) {
        var curItem = myFolder.item(i);
        if ( curItem.parentFolder != myFolder) {
            alert("error within AE: the parentFolder is not set correctly");
        }
        else {
            if ( !curItem.selected && curItem.typeName == "Footage") {
                //found an unselected solid.
                curItem.remove();
            }
        }
    }
}
}
```

**Attribut Item comment**

`app.project.item(index).comment`

**Description**

Chaîne contenant un commentaire, d'une taille maximale de 15 999 octets après toute conversion d'encodage. Ce commentaire est uniquement destiné à l'utilisateur ; il n'influe aucunement sur l'apparence et le comportement de l'élément.

**Type**

Chaîne ; lecture/écriture.

**Attribut Item id**

`app.project.item(index).id`

**Description**

Numéro d'identification unique et persistant utilisé en interne pour identifier un élément entre les sessions. La valeur de l'ID ne change pas lorsque le projet est enregistré dans un fichier et rechargé ultérieurement. Toutefois, lorsque vous importez ce projet dans un autre projet, de nouveaux ID sont attribués à tous les éléments du projet importé. L'ID ne s'affiche pas n'importe où dans l'interface utilisateur.

**Type**

Entier ; lecture seule.

**Attribut Item name**

`app.project.item(index).name`

**Description**

Nom de l'élément, tel qu'affiché dans le panneau Projet.

**Type**

Chaîne ; lecture/écriture.

**Attribut Item parentFolder**

`app.project.item(index).parentFolder`

**Description**

Objet FolderItem du dossier qui contient cet élément. Si cet élément se trouve au niveau supérieur du projet, il s'agit du dossier racine du projet (`app.project.rootFolder`). Vous pouvez utiliser la méthode `addFolder` d'ItemCollection pour ajouter un nouveau dossier et définir cette valeur pour placer des éléments dans le nouveau dossier. Voir la section « Méthode ItemCollection `addFolder()` », page 80.

**Type**

Objet FolderItem ; lecture/écriture.

**Exemple**

Le script suivant crée un nouvel objet FolderItem dans le panneau Projet et y place des compositions.

```
// create a new FolderItem in project, with name "comps"
var compFolder = app.project.items.addFolder("comps");
// move all compositions into new folder by setting
// compItem's parentFolder to "comps" folder
for(var i = 1; i <= app.project.numItems; i++) {
  if(app.project.item(i) instanceof CompItem)
    app.project.item(i).parentFolder = compFolder;
}
```

**Méthode Item remove()**

```
app.project.item(index).remove()
```

**Description**

Supprime cet élément du projet et du panneau Projet. Si l'élément est un FolderItem, tous les éléments contenus dans le dossier sont également supprimés du projet. Aucun fichier ou dossier n'est supprimé du disque.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Attribut Item selected**

```
app.project.item(index).selected
```

**Description**

Lorsque cet attribut est défini sur true, cet élément est sélectionné. Plusieurs éléments peuvent être sélectionnés simultanément. Définissez cet attribut sur true pour que le programme sélectionne l'élément ou sur false pour qu'il le désélectionne.

**Type**

Booléen ; lecture/écriture.

**Attribut Item typeName**

```
app.project.item(index).typeName
```

**Description**

Nom du type d'élément lisible par l'utilisateur ; par exemple : « Folder », « Footage » ou « Composition ».

**Type**

Chaîne ; lecture seule.

## Objet ItemCollection

`app.project.items`

### Description

L'objet `ItemCollection` représente une collection d'éléments. L'`ItemCollection` appartenant à un objet `Project` contient tous les objets `Item` correspondant aux éléments du projet. L'`ItemCollection` appartenant à un objet `FolderItem` contient tous les objets `Item` correspondant aux éléments de ce dossier.

- `ItemCollection` est une sous-classe de `Collection`. Toutes les méthodes et tous les attributs de `Collection`, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez `ItemCollection`. Voir la section « `Objet Collection` », page 51.

### Méthodes

Méthode	Référence	Description
<code>addComp()</code>	« Méthode <code>ItemCollection.addComp()</code> », page 79	Crée un nouvel objet <code>CompItem</code> et l'ajoute à la collection.
<code>addFolder()</code>	« Méthode <code>ItemCollection.addFolder()</code> », page 80	Crée un nouvel objet <code>FolderItem</code> et l'ajoute à la collection.

### Méthode `ItemCollection.addComp()`

`app.project.itemCollection.addComp(name, width, height, pixelAspect, duration, frameRate)`

#### Description

Crée une nouvelle composition. Crée et renvoie un nouvel objet `CompItem` et l'ajoute à cette collection.

Si l'`ItemCollection` appartient au projet ou au dossier racine, le `parentFolder` du nouvel élément est le dossier racine. Si l'`ItemCollection` appartient à n'importe quel autre dossier, le `parentFolder` du nouvel élément est ce `FolderItem`.

#### Paramètres

<code>name</code>	Chaîne contenant le nom de la composition.
<code>width</code>	Largeur de la composition, en pixels. Nombre entier compris dans la plage [4..30000].
<code>height</code>	Hauteur de la composition, en pixels. Nombre entier compris dans la plage [4..30000].
<code>pixelAspect</code>	Format des pixels de la composition. Valeur à virgule flottante comprise dans la plage [0.01..100.0].
<code>duration</code>	Durée de la composition, en secondes. Valeur à virgule flottante comprise dans la plage [0.0..10800.0].
<code>frameRate</code>	Cadence d'image de la composition. Valeur à virgule flottante comprise dans la plage [1.0..99.0]

#### Renvoi

Objet `CompItem`.

**Méthode ItemCollection addFolder()**

`app.project.itemCollection.addFolder(name)`

**Description**

Crée un nouveau dossier. Crée et renvoie un nouvel objet FolderItem et l'ajoute à cette collection.

Si l'ItemCollection appartient au projet ou au dossier racine, le `parentFolder` du nouveau dossier est le dossier racine. Si l'ItemCollection appartient à n'importe quel autre dossier, le `parentFolder` du nouveau dossier est ce FolderItem.

Pour placer des éléments dans le dossier, définissez l'attribut `parentFolder` de l'objet Item ; voir la section « Attribut Item `parentFolder` », page 77.

**Paramètres**

<code>name</code>	Chaîne contenant le nom du dossier.
-------------------	-------------------------------------

**Renvoi**

Objet FolderItem.

**Exemple**

Le script suivant crée un nouvel objet FolderItem dans le panneau Projet et y place des compositions.

```
// create a new FolderItem in project, with name "comps"
var compFolder = app.project.items.addFolder("comps");
// move all compositions into new folder by setting
// compItem's parentFolder to "comps" folder
for(var i = 1; i <= app.project.numItems; i++) {
  if(app.project.item(i) instanceof CompItem)
    app.project.item(i).parentFolder = compFolder;
}
```

## Objet KeyframeEase

```
myKey = new KeyframeEase(speed, influence);
```

### Description

L'objet KeyframeEase incorpore les paramètres de lissage d'image clé de la propriété AE d'un calque. Le lissage d'image clé est déterminé par les valeurs de vitesse et d'influence définies à l'aide de la méthode `setTemporalEaseAtKey` de la propriété. Voir la section « Méthode Property `setTemporalEaseAtKey()` », page 137.

Le constructeur crée un objet KeyframeEase. Les deux paramètres sont requis.

- `speed` : valeur à virgule flottante. Définit l'attribut `speed`.
- `influence` : valeur à virgule flottante comprise dans la plage [0.1..100.0]. Définit l'attribut `influence`.

### Exemple

L'exemple suivant suppose que la propriété spatiale `Position` possède plus de deux images clés.

```
var easeIn = new KeyframeEase(0.5, 50);
var easeOut = new KeyframeEase(0.75, 85);
var myPositionProperty = app.project.item(1).layer(1).property("Position")
myPositionProperty.setTemporalEaseAtKey(2, [easeIn], [easeOut]);
```

L'exemple suivant définit `Scale`, propriété temporelle à deux ou trois dimensions. Pour les propriétés 2D et 3D, vous devez définir une valeur de lissage à l'approche (`easeIn`) et une valeur de lissage à l'éloignement (`easeOut`) pour chaque dimension.

```
var easeIn = new KeyframeEase(0.5, 50);
var easeOut = new KeyframeEase(0.75, 85);
var myScaleProperty = app.project.item(1).layer(1).property("Scale")
myScaleProperty.setTemporalEaseAtKey(2, [easeIn, easeIn, easeIn], [easeOut, easeOut, easeOut]);
```

### Attributs

Attribut	Référence	Description
<code>speed</code>	« Attribut KeyframeEase <code>speed</code> », page 82	Paramètre de vitesse d'une image clé.
<code>influence</code>	« Attribut KeyframeEase <code>influence</code> », page 81	Paramètre d'influence d'une image clé.

### Attribut KeyframeEase `influence`

```
myKey.influence
```

### Description

Valeur de l'influence de l'image clé, comme indiqué dans la boîte de dialogue Vitesse d'image clé.

### Type

Valeur à virgule flottante comprise dans la plage [0.1..100.0] ; lecture/écriture.

**Attribut KeyframeEase speed**

*myKey.speed*

**Description**

Valeur de la vitesse de l'image clé. Les unités dépendent du type de l'image clé et s'affichent dans la boîte de dialogue Vitesse d'image clé.

**Type**

Valeur à virgule flottante ; lecture/écriture.

## Objet Layer

`app.project.item(index).layer(index)`

### Description

L'objet Layer fournit l'accès aux calques contenus dans des compositions. Il est accessible à partir de la collection de calques d'un élément, via un numéro d'index ou une chaîne de nom.

- Layer est la classe de base de CameraLayer, TextLayer, LightLayer et AVLayer. Ainsi, les attributs et les méthodes Layer sont disponibles lorsque vous utilisez les différents types de calque. Voir les sections « Objet AVLayer », page 39, « Objet CameraLayer », page 50, « Objet LightLayer », page 97 et « Objet TextLayer », page 174.

Les calques contiennent des propriétés AE, en plus de leurs attributs et méthodes JavaScript. Pour des exemples d'accès aux propriétés dans des calques, reportez-vous à la section « Objet PropertyBase », page 141.

### Exemple

Si le premier élément du projet est un CompItem, l'exemple suivant désactive le premier calque de cette composition et le renomme. Il peut par exemple permettre de désactiver une icône de la composition.

```
var firstLayer = app.project.item(1).layer(1);
firstLayer.enabled = false;
firstLayer.name = "Disabled Layer";
```

### Attributs

Attribut	Référence	Description
index	« Attribut Layer index », page 87	Position d'index du calque.
name	« Attribut Layer name », page 89	Nom du calque.
parent	« Attribut Layer parent », page 89	Parent de ce calque.
time	« Attribut Layer time », page 91	Instant courant du calque.
startTime	« Attribut Layer startTime », page 91	Instant de début du calque.
stretch	« Attribut Layer stretch », page 91	Extension temporelle du calque, en pourcentage.
inPoint	« Attribut Layer inPoint », page 87	Point d'entrée du calque.
outPoint	« Attribut Layer outPoint », page 89	Point de sortie du calque.
enabled	« Attribut Layer enabled », page 86	Lorsque cet attribut est défini sur true, le calque est activé.
solo	« Attribut Layer solo », page 91	Lorsque cet attribut est défini sur true, le calque est isolé.
shy	« Attribut Layer shy », page 90	Lorsque cet attribut est défini sur true, le calque est discret.
locked	« Attribut Layer locked », page 87	Lorsque cet attribut est défini sur true, le calque est verrouillé.
hasVideo	« Attribut Layer hasVideo », page 87	Lorsque cet attribut est défini sur true, le calque contient un composant vidéo.
active	« Attribut Layer active », page 84	Lorsque cet attribut est défini sur true, le calque est actif à l'instant courant.
nullLayer	« Attribut Layer nullLayer », page 89	Lorsque cet attribut est défini sur true, il s'agit d'un calque nul.
selectedProperties	« Attribut Layer selectedProperties », page 90	Toutes les propriétés AE sélectionnées dans le calque.
comment	« Attribut Layer comment », page 85	Commentaire descriptif pour le calque.

Attribut	Référence	Description
containingComp	« Attribut Layer containingComp », page 85	Composition contenant ce calque.
isNameSet	« Attribut Layer isNameSet », page 87	Lorsque cet attribut est défini sur true, le nom du calque a été explicitement défini.

### Méthodes

Méthode	Référence	Description
remove()	« Méthode Layer remove() », page 90	Supprime le calque de la composition.
moveToBeginning()	« Méthode Layer moveToBeginning() », page 88	Déplace le calque vers le haut de la composition (en position de premier calque).
moveToEnd()	« Méthode Layer moveToEnd() », page 88	Déplace le calque vers le bas de la composition (en position de dernier calque).
moveAfter()	« Méthode Layer moveAfter() », page 88	Déplace le calque sous un autre calque.
moveBefore()	« Méthode Layer moveBefore() », page 88	Déplace le calque au-dessus d'un autre calque.
duplicate()	« Méthode Layer duplicate() », page 86	Duplique le calque.
copyToComp()	« Méthode Layer copyToComp() », page 86	Copie le calque vers le haut (le début) d'une autre composition.
activeAtTime()	« Méthode Layer activeAtTime() », page 85	Indique si ce calque est actif à un instant donné.
setParentWithJump()	« Méthode Layer setParentWithJump() », page 90	Définit un nouveau parent pour ce calque.
applyPreset()	« Méthode Layer applyPreset() », page 85	Applique au calque une collection nommée de paramètres d'animation.

### Attribut Layer active

`app.project.item(index).layer(index).active`

#### Description

Lorsque cet attribut est défini sur true, le composant vidéo du calque est actif à l'instant courant.

Pour que sa valeur soit true, le calque doit être activé, aucun autre calque ne doit être isolé, sauf si ce calque l'est aussi, et l'instant donné doit être compris entre le point d'entrée (`inPoint`) et le point de sortie (`outPoint`) de ce calque.

Dans un calque audio, cette valeur n'est jamais true ; il existe un attribut `audioActive` distinct dans l'objet `AVLayer`.

#### Type

Booléen ; lecture seule.

**Méthode Layer activeAtTime()**

```
app.project.item(index).layer(index).activeAtTime(time)
```

**Description**

Renvoie true si ce calque est actif à l'instant spécifié. Pour que la valeur soit true, le calque doit être activé, aucun autre calque ne doit être isolé, sauf si ce calque l'est aussi, et l'instant donné doit être compris entre le point d'entrée (`inPoint`) et le point de sortie (`outPoint`) de ce calque.

**Paramètres**

time	Temps, en secondes. Valeur à virgule flottante.
------	---

**Renvoi**

Booléen.

**Méthode Layer applyPreset()**

```
app.project.item(index).layer(index).applyPreset(presetName);
```

**Description**

Applique au calque la collection spécifiée de paramètres d'animation (une animation prédéfinie). Les fichiers d'animation prédéfinie se trouvent dans le dossier Presets. Les utilisateurs peuvent créer de nouvelles animations prédéfinies via l'interface utilisateur.

**Paramètres**

presetName	Objet ExtendScript File du fichier contenant l'animation prédéfinie.
------------	--

**Renvoi**

Aucun.

**Attribut Layer comment**

```
app.project.item(index).layer(index).comment
```

**Description**

Commentaire descriptif pour le calque.

**Type**

Chaîne ; lecture/écriture.

**Attribut Layer containingComp**

```
app.project.item(index).layer(index).containingComp
```

**Description**

Composition contenant ce calque.

**Type**

Objet CompItem ; lecture seule.

**Méthode Layer copyToComp()**

```
app.project.item(index).layer(index).copyToComp(intoComp)
```

**Description**

Copie le calque dans la composition spécifiée. Le calque d'origine reste inchangé. Crée un nouvel objet Layer avec les mêmes valeurs que ce dernier, et place ce nouvel objet au début de la collection de calques (layers) dans l'objet CompItem cible. Extrayez la copie en utilisant `intoComp.layer(1)`.

Effectuer une copie dans un calque modifie les positions d'index des calques déjà présents dans la composition de destination. Cela équivaut à copier et coller un calque via l'interface utilisateur.

**Paramètres**

intoComp	Composition de destination et objet CompItem.
----------	---

**Renvoi**

Aucun.

**Méthode Layer duplicate()**

```
app.project.item(index).layer(index).duplicate()
```

**Description**

Duplique le calque. Crée un nouvel objet Layer dans lequel toutes les valeurs sont identiques à celles de ce calque. Elle produit le même résultat que la sélection d'un calque dans l'interface utilisateur suivie de la sélection de la commande Edition > Dupliquer, excepté que la sélection dans l'interface utilisateur ne change pas lorsque vous appelez cette méthode.

**Paramètres**

Aucun.

**Renvoi**

Objet Layer.

**Attribut Layer enabled**

```
app.project.item(index).layer(index).enabled
```

**Description**

Lorsque cet attribut est défini sur true, le calque est activé ; sinon, sa valeur est false. Correspond à l'état du bouton Vidéo correspondant au calque dans le panneau Montage.

**Type**

Booléen ; lecture/écriture.

**Attribut Layer hasVideo**

`app.project.item(index).layer(index).hasVideo`

**Description**

Lorsque cet attribut est défini sur true, le calque possède un bouton Vidéo (icône en forme d'œil) dans le panneau Montage ; sinon, sa valeur est false.

**Type**

Booléen ; lecture seule.

**Attribut Layer index**

`app.project.item(index).layer(index).index`

**Description**

Position d'index du calque.

**Type**

Nombre entier compris dans la plage [1..numLayers] ; lecture seule.

**Attribut Layer inPoint**

`app.project.item(index).layer(index).inPoint`

**Description**

Point d'entrée du calque, exprimé en secondes.

**Type**

Valeur à virgule flottante comprise dans la plage [-10800.0..10800.0] (+/- 3 heures) ; lecture/écriture.

**Attribut Layer isNameSet**

`app.project.item(index).layer(index).isNameSet`

**Description**

True si la valeur de l'attribut Name a été définie explicitement et non automatiquement à partir de la source.

**Type**

Booléen ; lecture seule.

**Attribut Layer locked**

`app.project.item(index).layer(index).locked`

**Description**

Lorsque cet attribut est défini sur true, le calque est verrouillé ; sinon, sa valeur est false. Correspond à la commande d'activation/de désactivation du verrouillage dans le panneau Calque.

**Type**

Booléen ; lecture/écriture.

**Méthode Layer moveAfter()**

```
app.project.item(index).layer(index).moveAfter(layer)
```

**Description**

Déplace ce calque à une position suivant immédiatement (au-dessous) le calque spécifié.

**Paramètres**

layer	Calque cible, objet Layer dans la même composition.
-------	---

**Renvoi**

Aucun.

**Méthode Layer moveBefore()**

```
app.project.item(index).layer(index).moveBefore(layer)
```

**Description**

Déplace ce calque à une position précédant immédiatement (au-dessus) le calque spécifié.

**Paramètres**

layer	Calque cible, objet Layer dans la même composition.
-------	---

**Renvoi**

Aucun.

**Méthode Layer moveToBeginning()**

```
app.project.item(index).layer(index).moveToBeginning()
```

**Description**

Déplace ce calque à la toute première position de la pile de calques (premier calque).

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Méthode Layer moveToEnd()**

```
app.project.item(index).layer(index).moveToEnd()
```

**Description**

Déplace ce calque à la toute dernière position de la pile de calques (dernier calque).

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Attribut Layer name**

```
app.project.item(index).layer(index).name
```

**Description**

Nom du calque. Par défaut, il s'agit du même nom que celui de la source (lequel ne peut pas être modifié dans le panneau Calque), mais vous pouvez le modifier.

**Type**

Chaîne ; lecture/écriture.

**Attribut Layer nullLayer**

```
app.project.item(index).layer(index).nullLayer
```

**Description**

Lorsque cet attribut est défini sur true, le calque a été créé en tant qu'objet nul ; sinon, sa valeur est false.

**Type**

Booléen ; lecture seule.

**Attribut Layer outPoint**

```
app.project.item(index).layer(index).outPoint
```

**Description**

Point de sortie du calque, exprimé en secondes.

**Type**

Valeur à virgule flottante comprise dans la plage [-10800.0..10800.0] (+/- 3 heures) ; lecture/écriture.

**Attribut Layer parent**

```
app.project.item(index).layer(index).parent
```

**Description**

Parent de ce calque ; peut être nul.

Les valeurs de décalage sont calculées de manière à compenser toute transformation supérieure dans la hiérarchie. Ainsi, lorsque vous définissez le parent, il n'y a aucun déplacement apparent dans la transformation du calque. Si, par exemple, la rotation du nouveau parent est de 30 degrés, une rotation de -30 degrés est appliquée au calque enfant.

Pour définir le parent sans modifier les valeurs de transformation du calque enfant, utilisez la méthode `setParentWithJump`.

**Type**

Objet Layer ou valeur nulle ; lecture/écriture.

**Méthode Layer remove()**

```
app.project.item(index).layer(index).remove()
```

**Description**

Supprime le calque spécifié de la composition.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Attribut Layer selectedProperties**

```
app.project.item(index).layer(index).selectedProperties
```

**Description**

Tableau contenant tous les objets Property et PropertyGroup actuellement sélectionnés dans le calque.

**Type**

Tableau d'objets PropertyBase ; lecture seule.

**Méthode Layer setParentWithJump()**

```
app.project.item(index).layer(index).setParentWithJump(newParent)
```

**Description**

Définit le parent de ce calque sur le calque spécifié, sans modifier les valeurs de transformation du calque enfant. Un déplacement de rotation, de translation ou de mise à l'échelle du calque enfant peut être apparent, car les valeurs de transformation de ce calque sont combinées à celles de ses prédécesseurs.

Si vous ne souhaitez pas que le calque enfant se déplace, définissez l'attribut `parent` directement. Dans ce cas, un décalage est calculé et défini dans les trames de transformation du calque enfant afin d'empêcher le déplacement de se produire.

**Paramètres**

<code>newParent</code>	Objet Layer dans la même composition.
------------------------	---------------------------------------

**Renvoi**

Aucun.

**Attribut Layer shy**

```
app.project.item(index).layer(index).shy
```

**Description**

Lorsque cet attribut est défini sur `true`, le calque est discret, ce qui signifie qu'il est masqué dans le panneau Calque si l'option Masquer calques discrets de la composition est activée.

**Type**

Booléen ; lecture/écriture.

**Attribut Layer solo**

```
app.project.item(index).layer(index).solo
```

**Description**

Lorsque cet attribut est défini sur true, le calque est isolé ; sinon, sa valeur est false.

**Type**

Booléen ; lecture/écriture.

**Attribut Layer startTime**

```
app.project.item(index).layer(index).startTime
```

**Description**

Heure de début du calque, exprimée en secondes.

**Type**

Valeur à virgule flottante comprise dans la plage [-10800.0..10800.0] (+/- 3 heures) ; lecture/écriture.

**Attribut Layer stretch**

```
app.project.item(index).layer(index).stretch
```

**Description**

Extension temporelle du calque, exprimée en pourcentage. Une valeur de 100 n'applique aucune extension. Les valeurs comprises entre 0 et 1 sont définies sur 1. Les valeurs comprises entre -1 et 0 (0 exclus) sont définies sur -1.

**Type**

Valeur à virgule flottante comprise dans la plage [-9900.0..9900.0] ; lecture/écriture.

**Attribut Layer time**

```
app.project.item(index).layer(index).time
```

**Description**

Instant courant du calque, exprimé en secondes.

**Type**

Valeur à virgule flottante ; lecture seule.

## Objet LayerCollection

`app.project.item(index).layers`

### Description

L'objet LayerCollection représente un ensemble de calques. La LayerCollection appartenant à un objet CompItem contient tous les objets Layer pour les calques de la composition. Les méthodes de l'objet Collection vous permettent de manipuler la liste des calques.

- LayerCollection est une sous-classe de Collection. Toutes les méthodes et tous les attributs de Collection, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez LayerCollection. Voir la section « Objet Collection », page 51.

### Exemple

Partant du principe que le premier élément du projet est un CompItem et que le second est un AVItem, le code suivant permet d'afficher le nombre de calques dans la collection de calques de CompItem, d'ajouter un nouveau calque basé sur un AVItem du projet et d'afficher le nouveau nombre de calques.

```
var firstComp = app.project.item(1);
var layerCollection = firstComp.layers;
alert("number of layers before is " + layerCollection.length);
var anAVItem = app.project.item(2);
layerCollection.add(anAVItem);
alert("number of layers after is " + layerCollection.length);
```

### Méthodes

Méthode	Référence	Description
<code>add()</code>	« Méthode LayerCollection add() », page 93	Crée un nouveau AVLayer et l'ajoute à cette collection.
<code>addNull()</code>	« Méthode LayerCollection addNull() », page 94	Crée un nouveau calque nul et l'ajoute à cette collection.
<code>addSolid()</code>	« Méthode LayerCollection addSolid() », page 94	Crée un nouveau calque, un FootageItem avec un SolidSource, et l'ajoute à cette collection.
<code>addText()</code>	« Méthode LayerCollection addText() », page 95	Crée un nouveau calque de texte et l'ajoute à cette collection.
<code>addCamera()</code>	« Méthode LayerCollection addCamera() », page 93	Crée un nouveau calque de caméra et l'ajoute à cette collection.
<code>addLight()</code>	« Méthode LayerCollection addLight() », page 93	Crée un nouveau calque de lumière et l'ajoute à cette collection.
<code>addShape()</code>	« Méthode LayerCollection addShape() », page 94	Crée un nouveau calque de forme et l'ajoute à cette collection.
<code>byName()</code>	« Méthode LayerCollection byName() », page 95	Extrait l'objet calque portant le nom spécifié.
<code>precompose()</code>	« Méthode LayerCollection precompose() », page 96	Rassemble les calques spécifiés dans une nouvelle composition.

**Méthode LayerCollection add()**

```
app.project.item(index).layers.add(item, duration)
```

**Description**

Crée un nouvel objet AVLayer contenant l'élément spécifié et l'ajoute à cette collection.

Cette méthode génère une exception si l'élément ne peut pas être ajouté en tant que calque à cette composition.

**Paramètres**

item	Objet AVItem de l'élément à ajouter.
duration	Facultatif, durée en secondes d'un calque fixe, valeur à virgule flottante. Utilisé uniquement si l'élément contient une partie de métrage fixe. N'affecte ni les films, ni les séquences, ni l'audio.  Défini, ce paramètre détermine la durée du nouveau calque. Sinon, la durée est définie en fonction des préférences utilisateur. Par défaut, la durée est identique à celle du Compltem. Pour spécifier une valeur préférée, choisissez Edition > Préférences > Importer (Windows) ou After Effects > Préférences > Importer (Mac OS), puis sélectionnez les options désirées dans la zone Métrage fixe.

**Renvoi**

Objet AVLayer.

**Méthode LayerCollection addCamera()**

```
app.project.item(index).layers.addCamera(name, centerPoint)
```

**Description**

Crée un nouveau calque de caméra et ajoute l'objet CameraLayer à cette collection.

**Paramètres**

name	Chaîne contenant le nom du nouveau calque.
centerPoint	Centre de la nouvelle caméra, tableau de valeurs à virgule flottante [x, y]. Permet de définir les valeurs initiales x et y de la propriété Point ciblé de la nouvelle caméra. La valeur z est égale à 0.

**Renvoi**

Objet CameraLayer.

**Méthode LayerCollection addLight()**

```
app.project.item(index).layers.addLight(name, centerPoint)
```

**Description**

Crée un nouveau calque de lumière et ajoute l'objet LightLayer à cette collection.

**Paramètres**

name	Chaîne contenant le nom du nouveau calque.
centerPoint	Centre de la nouvelle lumière, tableau de valeurs à virgule flottante [x, y].

**Renvoi**

Objet LightLayer.

**Méthode LayerCollection addNull()**

```
app.project.item(index).layers.addNull(duration)
```

**Description**

Crée un nouveau calque nul et ajoute l'objet AVLayer à cette collection. Cela revient à sélectionner la commande Calque > Créer > Objet nul.

**Paramètres**

duration	Facultatif, durée en secondes d'un calque fixe, valeur à virgule flottante.  Défini, ce paramètre détermine la durée du nouveau calque. Sinon, la durée est définie en fonction des préférences utilisateur. Par défaut, la durée est identique à celle du Compltem. Pour spécifier une valeur préférée, choisissez Edition > Préférences > Importer (Windows) ou After Effects > Préférences > Importer (Mac OS), puis sélectionnez les options désirées dans la zone Métrage fixe.
----------	--

**Renvoi**

Objet AVLayer.

**Méthode LayerCollection addShape()**

```
app.project.item(index).layers.addShape()
```

**Description**

Crée un nouvel objet ShapeLayer pour un nouveau calque de forme vide. Utilisez l'objet ShapeLayer pour ajouter des propriétés, telles qu'une forme, un fond, un contour et des filtres de contour.

Cela revient au même que d'utiliser un outil de forme en mode « L'outil crée une forme ». Les outils ajoutent automatiquement un groupe de vecteurs incluant un fond et un contour définis dans les options des outils.

**Paramètres**

Aucun.

**Renvoi**

Objet ShapeLayer.

**Méthode LayerCollection addSolid()**

```
app.project.item(index).layers.addSolid(color, name, width, height, pixelAspect, duration)
```

**Description**

Crée un nouvel objet SolidSource, avec les valeurs spécifiées ; définit le nouveau SolidSource comme la valeur mainSource d'un nouvel objet FootageItem et ajoute le FootageItem au projet. Crée un nouvel objet AVLayer, définit le nouveau FootageItem comme source et ajoute le calque à cette collection.

**Paramètres**

color	Couleur du solide, tableau de quatre valeurs à virgule flottante [R, V, B, A], comprises dans la plage [0.0..1.0].
name	Chaîne contenant le nom du solide.
width	Largeur du solide, en pixels. Nombre entier compris dans la plage [4..30000].

height	Hauteur du solide, en pixels. Nombre entier compris dans la plage [4..30000].
pixelAspect	Format des pixels du solide. Valeur à virgule flottante comprise dans la plage [0.01..100.0].
duration	Facultatif, durée en secondes d'un calque fixe, valeur à virgule flottante.  Défini, ce paramètre détermine la durée du nouveau calque. Sinon, la durée est définie en fonction des préférences utilisateur. Par défaut, la durée est identique à celle du Compltem. Pour spécifier une valeur préférée, choisissez Edition > Préférences > Importer (Windows) ou After Effects > Préférences > Importer (Mac OS), puis sélectionnez les options désirées dans la zone Métrage fixe.

**Renvoi**

Objet AVLayer.

**Méthode LayerCollection addText()**

```
app.project.item(index).layers.addText(sourceText)
```

**Description**

Crée un nouveau calque de texte et ajoute le nouvel objet TextLayer à cette collection.

**Paramètres**

sourceText	Facultatif ; chaîne contenant le texte source du nouveau calque ou objet TextDocument contenant le texte source du nouveau calque. Voir la section « Objet TextDocument », page 173.
------------	--

**Renvoi**

Objet TextLayer.

**Méthode LayerCollection byName()**

```
app.project.item(index).layers.byName(name)
```

**Description**

Renvoie le premier calque (calque supérieur) trouvé dans cette collection avec le nom spécifié, ou une valeur nulle si aucun calque portant le nom spécifié n'a été trouvé.

**Paramètres**

name	Chaîne contenant le nom.
------	--------------------------

**Renvoi**

Objet Layer ou valeur nulle.

**Méthode LayerCollection precompose()**

```
app.project.item(index).layers.precompose(layerIndices, name, moveAllAttributes)
```

**Description**

Crée un nouvel objet CompItem et déplace les calques spécifiés dans sa collection de calques. Les calques individuels sont alors supprimés de cette collection et le nouveau CompItem y est ajouté.

**Paramètres**

<code>layerIndices</code>	Index de position des calques à récupérer. Tableau d'entiers.
<code>name</code>	Nom du nouvel objet CompItem.
<code>moveAllAttributes</code>	Facultatif. Si la valeur est true (par défaut), tous les attributs de la nouvelle composition sont conservés. Cela revient à sélectionner l'option « Transférer tous les attributs vers la nouvelle composition » de la boîte de dialogue Précomposition.  Vous pouvez définir cette valeur sur false uniquement s'il n'existe qu'un seul index dans le tableau <code>layerIndices</code> . Cela revient à sélectionner l'option « Laisser tous les attributs dans » de la boîte de dialogue Précomposition.

**Renvoi**

Objet CompItem.

## Objet LightLayer

`app.project.item(index).layer(index)`

### Description

L'objet LightLayer représente un calque de lumière dans une composition. Vous pouvez le créer à l'aide de la méthode `addLight` de l'objet `LayerCollection` ; voir la section « Méthode `LayerCollection.addLight()` », page 93. Pour accéder à cet objet dans la collection de calques d'un élément, vous pouvez utiliser son numéro d'index ou encore une chaîne de nom.

- LightLayer est une sous-classe de Layer. Toutes les méthodes et tous les attributs de Layer sont disponibles lorsque vous utilisez LightLayer. Voir la section « Objet Layer », page 83.

### Propriétés AE

LightLayer ne définit pas d'attribut supplémentaire, mais possède des propriétés AE qui diffèrent des autres types de calques. Il dispose des propriétés et des groupes de propriétés suivants :

Marker

Transform

Point of Interest

Position

Scale

Orientation

X Rotation

Y Rotation

Rotation

Opacity

Light Options

Intensity

Color

Cone Angle

Cone Feather

Casts Shadows

Shadow Darkness

Shadow Diffusion

## Objet MarkerValue

`new MarkerValue(comment, chapter, url, frameTarget, cuePointName, params)`

### Description

L'objet `MarkerValue` représente un repère de calque qui associe un commentaire, et éventuellement un point de référence d'un chapitre, un lien vers une page Web ou un signal de départ vidéo Flash, à un point particulier d'un calque. Vous le créez à l'aide du constructeur ; tous les arguments à l'exception de `comment` sont facultatifs. Tous les arguments sont des chaînes définies dans les attributs correspondants de l'objet `MarkerValue` renvoyé, sauf `params`. Il s'agit d'un tableau comprenant des paires de valeurs clés, accessibles ensuite par le biais des méthodes `getParameters()` et `setParameters()`. Un script peut définir un nombre indifférent de paires de paramètres ; leur ordre ne reflète pas l'ordre d'affichage dans l'application.

Pour associer un repère à un calque, définissez l'objet `MarkerValue` dans la propriété `AE Marker` du calque.

```
layerObject.property("Marker").setValueAtTime(time, markerValueObject);
```

Pour plus d'informations sur l'utilisation des repères, consultez la rubrique « Utilisation des repères » dans l'aide d'After Effects.

### Attributs

Attribut	Référence	Description
<code>comment</code>	« Attribut <code>MarkerValue comment</code> », page 99	Commentaire sur le calque associé.
<code>chapter</code>	« Attribut <code>MarkerValue chapter</code> », page 99	Point de référence d'un lien de chapitre pour le calque associé.
<code>cuePointName</code>	« Attribut <code>MarkerValue cuePointName</code> », page 99	Nom du signal de départ vidéo Flash.
<code>eventCuePoint</code>	« Attribut <code>MarkerValue eventCuePoint</code> », page 99	Indique si le signal de départ vidéo Flash s'applique à un événement ou à la navigation.
<code>url</code>	« Attribut <code>MarkerValue url</code> », page 101	URL de la page Web à associer au calque.
<code>frameTarget</code>	« Attribut <code>MarkerValue frameTarget</code> », page 99	Image cible sur la page Web spécifiée par <code>url</code> .

### Méthodes

Méthode	Référence	Description
<code>getParameters()</code>	« Méthode <code>MarkerValue getParameters()</code> », page 100	Extrait les paires de valeurs clés associées à la valeur du repère.
<code>setParameters()</code>	« Méthode <code>MarkerValue setParameters()</code> », page 100	Définit les paires de valeurs clés associées à la valeur du repère.

### Exemples

- Pour définir un repère indiquant « Fade Up » au repère 2 secondes :

```
var myMarker = new MarkerValue("Fade Up");
myLayer.property("Marker").setValueAtTime(2, myMarker);
```

- Pour obtenir des valeurs comment d'un repère particulier :

```
var commentOfFirstMarker = app.project.item(1).layer(1).property("Marker").keyValue(1).comment;
var commentOfMarkerAtTime4 =
    app.project.item(1).layer(1).property("Marker").valueAtTime(4.0,true).comment
var markerProperty = app.project.item(1).layer(1).property("Marker");
var markerValueAtTimeClosestToTime4 =
    markerProperty.keyValue(markerProperty.nearestKeyIndex(4.0));
var commentOfMarkerClosestToTime4 = markerValueAtTimeClosestToTime4.comment;
```

**Attribut MarkerValue chapter**

```
app.project.item(index).layer(index).property("Marker").keyValue(index).chapter
```

**Description**

Lien texte de chapitre pour ce repère. Les liens de chapitre déclenchent un saut vers un chapitre d'un film QuickTime ou d'un autre format prenant en charge les repères de chapitre.

**Type**

Chaîne ; lecture/écriture.

**Attribut MarkerValue comment**

```
app.project.item(index).layer(index).property("Marker").keyValue(index).comment
```

**Description**

Texte de commentaire pour ce repère. Ce commentaire apparaît dans le panneau Montage en regard du repère de calque.

**Type**

Chaîne ; lecture/écriture.

**Attribut MarkerValue cuePointName**

```
app.project.item(index).layer(index).property("Marker").keyValue(index).cuePointName
```

**Description**

Nom du signal de départ vidéo Flash tel qu'il apparaît dans la boîte de dialogue Repère.

**Type**

Chaîne ; lecture/écriture.

**Attribut MarkerValue eventCuePoint**

```
app.project.item(index).layer(index).property("Marker").keyValue(index).eventCuePoint
```

**Description**

Lorsque cet attribut est `true`, le signal de départ vidéo Flash s'applique à un événement ; dans le cas contraire, il s'applique à la navigation.

**Type**

Booléen ; lecture/écriture.

**Attribut MarkerValue frameTarget**

```
app.project.item(index).layer(index).property("Marker").keyValue(index).frameTarget
```

**Description**

Image texte cible pour ce repère. Comme la valeur URL, cet attribut cible une image spécifique contenue dans une page Web.

**Type**

Chaîne ; lecture/écriture.

**Méthode MarkerValue getParameters()**

```
app.project.item(index).layer(index).property("Marker").keyValue(index).getParameters()
```

**Description**

Renvoie les paires de valeurs clés pour les paramètres de signal de départ vidéo Flash, pour un signal de départ associé à cette valeur de repère.

**Paramètres**

Aucun.

**Renvoi**

Un objet avec un attribut correspondant au nom de chaque paramètre et contenant la valeur de ce paramètre.

**Méthode MarkerValue setParameters()**

```
app.project.item(index).layer(index).property("Marker").keyValue(index).setParameters(keyValuePair)
```

**Description**

Associe un ensemble de paires de valeurs clés pour les paramètres de signal de départ vidéo Flash, pour un signal de départ associé à cette valeur de repère. Un signal de départ peut contenir un nombre indifférent de paramètres, mais vous pouvez seulement en ajouter trois via l'interface utilisateur ; utilisez cette méthode pour ajouter plus de trois paramètres.

**Paramètres**

keyValuePair	Objet contenant les paires de valeurs clés en tant qu'attributs et valeurs. La méthode toString() de l'objet est appelée pour affecter la valeur de chaîne de chaque attribut à la clé nommée.
--------------	--

**Renvoi**

Aucun.

**Exemple**

```
var mv = new MarkerValue("My Marker");

var parms = new Object;
parms.timeToBlink = 1;
parms.assignMe = "A string"

mv.setParameters(parms);

myLayer.property("Marker").setValueAtTime(2, mv);
```

**Attribut MarkerValue url**

```
app.project.item(index).layer(index).property("Marker").keyValue(index).url
```

**Description**

URL de ce repère. Cette URL est un lien automatique vers une page Web.

**Type**

Chaîne ; lecture/écriture.

## Objet MaskPropertyGroup

`app.project.item(index).layer(index).mask`

### Description

L'objet MaskPropertyGroup insère les attributs de masque dans un calque.

- MaskPropertyGroup est une sous-classe de PropertyGroup. Toutes les méthodes et tous les attributs de PropertyBase et PropertyGroup, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez MaskPropertyGroup. Voir les sections « Objet PropertyBase », page 141 et « Objet PropertyGroup », page 148.

### Attributs

Attribut	Référence	Description
maskMode	« Attribut MaskPropertyGroup maskMode », page 103	Mode masque.
inverted	« Attribut MaskPropertyGroup inverted », page 102	Lorsque cet attribut est true, le masque est inversé.
rotoBezier	« Attribut MaskPropertyGroup rotoBezier », page 104	Lorsque cet attribut est true, la forme du masque est RotoBézier.
maskMotionBlur	« Attribut MaskPropertyGroup maskMotionBlur », page 103	Manière dont le flou directionnel est appliqué à ce masque.
locked	« Attribut MaskPropertyGroup locked », page 103	Lorsque cet attribut est true, le masque est verrouillé.
color	« Attribut MaskPropertyGroup color », page 102	Couleur utilisée pour dessiner le contour du masque dans l'interface utilisateur.

### Attribut MaskPropertyGroup color

`app.project.item(index).layer(index).mask(index).color`

### Description

Couleur utilisée pour dessiner le contour du masque, tel qu'il apparaît dans l'interface utilisateur (panneau Composition, panneau Calque et panneau Montage).

### Type

Tableau de trois valeurs à virgule flottante, [R, V, B], comprises dans la plage [0.0..1.0] ; lecture/écriture.

### Attribut MaskPropertyGroup inverted

`app.project.item(index).layer(index).mask(index).inverted`

### Description

Lorsque cet attribut est true, le masque est inversé. Dans le cas contraire, cet attribut est false.

### Type

Booléen ; lecture/écriture.

**Attribut MaskPropertyGroup locked**

`app.project.item(index).layer(index).mask(index).locked`

**Description**

Lorsque cet attribut est true, le masque est verrouillé et ne peut être édité dans l'interface utilisateur. Dans le cas contraire, cet attribut est false.

**Type**

Booléen ; lecture/écriture.

**Attribut MaskPropertyGroup maskMode**

`app.project.item(index).layer(index).mask(index).maskMode`

**Description**

Mode de masque pour ce masque.

**Type**

Valeur énumérée MaskMode ; écriture/lecture. L'une des suivantes :

MaskMode.NONE

MaskMode.ADD

MaskMode.SUBTRACT

MaskMode.INTERSECT

MaskMode.LIGHTEN

MaskMode.DARKEN

MaskMode.DIFFERENCE

**Attribut MaskPropertyGroup maskMotionBlur**

`app.project.item(index).layer(index).mask(index).maskMotionBlur`

**Description**

Manière dont le flou directionnel est appliqué à ce masque.

**Type**

Valeur énumérée MaskMotionBlur ; lecture/écriture. L'une des suivantes :

MaskMotionBlur.SAME\_AS\_LAYER

MaskMotionBlur.ON

MaskMotionBlur.OFF

**Attribut MaskPropertyGroup rotoBezier**

`app.project.item(index).layer(index).mask(index).rotoBezier`

**Description**

Lorsque cet attribut est true, la forme du masque est RotoBézier. Dans le cas contraire, cet attribut est false.

**Type**

Booléen ; lecture/écriture.

## Objet OMCollection

`app.project.renderQueue.items.outputModules`

### Description

L'objet OMCollection contient tous les modules de sortie de la file d'attente de rendu. La collection permet d'accéder aux objets OutputModule, mais n'offre pas d'autres fonctionnalités. Le premier objet OutputModule de la collection se trouve à la position d'index 1. Voir la section « [Objet OutputModule](#) », page 106.

- OMCollection est une sous-classe de Collection. Toutes les méthodes et tous les attributs de Collection sont disponibles lorsque vous utilisez OMCollection. Voir la section « [Objet Collection](#) », page 51.

## Objet OutputModule

`app.project.renderQueue.item(index).outputModules(index)`

### Description

Un objet OutputModule de renderQueueItem génère un fichier ou une séquence unique par le biais d'un rendu, et contient les attributs et méthodes associés au fichier pour lequel le rendu doit être effectué.

### Attributs

Attribut	Référence	Description
file	« Attribut OutputModule file », page 106	Chemin et nom du fichier dont le rendu doit être effectué.
postRenderAction	« Attribut OutputModule postRenderAction », page 107	Action à effectuer avant le rendu.
name	« Attribut OutputModule name », page 107	Nom de l'interface utilisateur du module de sortie.
templates	« Attribut OutputModule templates », page 108	Tous les modèles du module de sortie.

### Méthodes

Méthode	Référence	Description
remove()	« Méthode OutputModule remove() », page 107	Supprime le module de sortie de la liste des éléments de la file d'attente de rendu.
saveAsTemplate()	« Méthode OutputModule saveAsTemplate() », page 107	Enregistre un nouveau modèle de module de sortie.
applyTemplate()	« Méthode OutputModule applyTemplate() », page 106	Applique un modèle de module de sortie.

### Méthode OutputModule applyTemplate()

`app.project.renderQueue.item(index).outputModules[index].applyTemplate(templateName)`

### Description

Applique le modèle de module de sortie prédéfini existant.

### Paramètres

templateName	Chaîne contenant le nom du modèle à appliquer.
--------------	--

### Renvoi

Aucun.

### Attribut OutputModule file

`app.project.renderQueue.item(index).outputModules[index].file`

### Description

Objet ExtendScript File du fichier dont le rendu doit être effectué avec ce module de sortie.

### Type

Objet ExtendScript File ; lecture/écriture.

**Attribut OutputModule name**

```
app.project.renderQueue.item(index).outputModules[index].name
```

**Description**

Nom du module de sortie, tel qu'il apparaît dans l'interface utilisateur.

**Type**

Chaîne ; lecture seule.

**Attribut OutputModule postRenderAction**

```
app.project.renderQueue.item(index).outputModules[index].postRenderAction
```

**Description**

Action à effectuer lorsque l'opération de rendu est terminée.

**Type**

Valeur énumérée PostRenderAction (lecture/écriture) ; l'une des suivantes :

```
postRenderAction.NONE  
postRenderAction.IMPORT  
postRenderAction.IMPORT_AND_REPLACE_USAGE  
postRenderAction.SET_PROXY
```

**Méthode OutputModule remove()**

```
app.project.renderQueue.item(index).outputModules[index].remove()
```

**Description**

Supprime cet objet OutputModule de la collection.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Méthode OutputModule saveAsTemplate()**

```
app.project.renderQueue.item(index).outputModules[index].saveAsTemplate(name)
```

**Description**

Enregistre ce module de sortie comme modèle et l'ajoute au tableau des modèles (templates).

**Paramètres**

name	Chaîne contenant le nom du nouveau modèle.
------	--

**Renvoi**

Aucun.

**Attribut OutputModule templates**

`app.project.renderQueue.item(index).outputModules[index].templates`

**Description**

Noms de tous les modèles de modules de sortie disponibles dans l'installation locale d'After Effects.

**Type**

Tableau de chaînes ; lecture seule.

## Objet PlaceholderSource

```
app.project.item(index).mainSource  
app.project.item(index).proxySource
```

### Description

L'objet PlaceholderSource décrit la source du métrage d'un pseudo-élément.

PlaceholderSource est une sous-classe de FootageSource. Toutes les méthodes et tous les attributs de FootageSource sont disponibles lorsque vous utilisez PlaceholderSource. Voir la section « Objet FootageSource », page 67. PlaceholderSource ne définit pas d'attributs ou de méthodes supplémentaires.

## Objet Project

app.project

### Description

L'objet Project représente un projet After Effects. Les attributs donnent non seulement accès à des objets spécifiques au sein du projet, tels que fichiers ou métrages importés et compositions, mais aussi aux paramètres de projet tels que la base du code temporel. Les méthodes permettent d'importer le métrage, de créer des solides, des compositions et des dossiers, et d'enregistrer les modifications.

### Attributs

Attribut	Référence	Description
file	« Attribut Project file », page 113	Fichier du projet actuellement ouvert.
rootFolder	« Attribut Project rootFolder », page 116	Dossier contenant tous les commentaires du projet ; équivalent du panneau Projet.
items	« Attribut Project items », page 114	Tous les éléments du projet.
activeItem	« Attribut Project activeItem », page 111	Élément actuellement actif.
bitsPerChannel	« Attribut Project bitsPerChannel », page 112	Codage des couleurs du projet en cours.
transparencyGridThumbnails	« Attribut Project transparencyGridThumbnails », page 118	Lorsque cet attribut est défini sur true, les vignettes utilisent le motif de damier de transparence.
timecodeDisplayType	« Attribut Project timecodeDisplayType », page 118	Mode d'affichage du code temporel.
timecodeBaseType	« Attribut Project timecodeBaseType », page 117	Paramètre de base du code temporel du projet.
timecodeNTSCDropFrame	« Attribut Project timecodeNTSCDropFrame », page 118	Paramètre d'image compensée du projet.
timecodeFilmType	« Attribut Project timecodeFilmType », page 118	Type de film pour le paramètre du projet « Pieds + images ».
numItems	« Attribut Project numItems », page 115	Nombre total d'éléments contenus dans le projet.
selection	« Attribut Project selection », page 117	Tous les éléments sélectionnés dans le panneau Projet.
renderQueue	« Attribut Project renderQueue », page 116	File d'attente de rendu du projet.
displayStartFrame	« Attribut Project displayStartFrame », page 113	Image à partir de laquelle commence la numérotation lors de l'affichage du projet.
linearBlending	« Attribut Project linearBlending », page 114	Lorsque cet attribut est défini sur true, la fusion linéaire est utilisée dans le projet.

### Méthodes

Méthode	Référence	Description
item()	« Méthode Project item() », page 114	Extrait un élément du projet.
consolidateFootage()	« Méthode Project consolidateFootage() », page 112	Consolide tout le métrage du projet.

Méthode	Référence	Description
<code>removeUnusedFootage()</code>	« Méthode Project <code>removeUnusedFootage()</code> », page 115	Supprime tout métrage non utilisé du projet.
<code>reduceProject()</code>	« Méthode Project <code>reduceProject()</code> », page 115	Réduit le projet à un ensemble donné d'éléments.
<code>close()</code>	« Méthode Project <code>close()</code> », page 112	Ferme le projet avec les options d'enregistrement habituelles.
<code>save()</code>	« Méthode Project <code>save()</code> », page 116	Enregistre le projet.
<code>saveWithDialog()</code>	« Méthode Project <code>saveWithDialog()</code> », page 116	Affiche une boîte de dialogue Enregistrer.
<code>importPlaceholder()</code>	« Méthode Project <code>importFileWithDialog()</code> », page 113	Importe un pseudo-élément dans le projet.
<code>importFile()</code>	« Méthode Project <code>importFile()</code> », page 113	Importe un fichier dans le projet.
<code>importFileWithDialog()</code>	« Méthode Project <code>importFileWithDialog()</code> », page 113	Affiche une boîte de dialogue Importer fichier.
<code>showWindow()</code>	« Méthode Project <code>showWindow()</code> », page 117	Affiche ou masque le panneau Projet.
<code>autoFixExpressions()</code>	« Méthode Project <code>autoFixExpressions()</code> », page 111	Remplace automatiquement le texte dans toutes les expressions.

### Attribut Project `activeItem`

`app.project.activeItem`

#### Description

Élément actuellement actif et devant être soumis à une action, ou valeur nulle si aucun élément n'est actuellement sélectionné ou si plusieurs éléments sont sélectionnés.

#### Type

Objet `Item` ou nul ; lecture seule.

### Méthode Project `autoFixExpressions()`

`app.project.autoFixExpressions(oldText, newText)`

#### Description

Remplace automatiquement le texte des expressions rompues du projet, si le nouveau texte provoque l'évaluation sans erreur de l'expression.

#### Paramètres

<code>oldText</code>	Texte à remplacer.
<code>newText</code>	Nouveau texte.

#### Renvoi

Aucun.

**Attribut Project bitsPerChannel**

app.project.bitsPerChannel

**Description**

Codage des couleurs du projet en cours, soit 8, 16 ou 32 bits.

**Type**

Entier (8, 16 ou 32 uniquement) ; lecture/écriture.

**Méthode Project close()**

app.project.close(*closeOptions*)

**Description**

Ferme le projet avec possibilité de sauvegarder les modifications automatiquement, en invitant l'utilisateur à sauvegarder les modifications ou à fermer sans sauvegarder.

**Paramètres**

CloseOptions	Action à exécuter à la fermeture. Valeur énumérée CloseOptions ; l'une des suivantes : CloseOptions.DO_NOT_SAVE_CHANGES : ferme sans enregistrer. CloseOptions.PROMPT_TO_SAVE_CHANGES : demande de confirmer l'enregistrement des modifications avant la fermeture. CloseOptions.SAVE_CHANGES : enregistre automatiquement à la fermeture.
--------------	---

**Renvoi**

Booléen. True une fois l'action exécutée. False si le fichier n'a pas été enregistré auparavant, qu'une boîte de dialogue d'enregistrement s'affiche et que l'utilisateur annule l'enregistrement.

**Méthode Project consolidateFootage()**

app.project.consolidateFootage()

**Description**

Consolide tout le métrage du projet. Equivalent de la commande Fichier > Consolider tout le métrage.

**Paramètres**

Aucun.

**Renvoi**

Entier ; nombre total d'éléments de métrage supprimés.

**Attribut Project displayStartFrame**`app.project.displayStartFrame`**Description**

Image à partir de laquelle commence la numérotation lors de l'affichage du projet avec une valeur `timecodeDisplayType` de `TimecodeDisplayType.FRAMES`. (Voir la section « Attribut Project `timecodeDisplayType` », page 118.) Equivalent au paramètre « Commencer à numérotter les images à : » dans Paramètres du projet > Style d'affichage.

**Type**

Entier ; lecture/écriture.

**Attribut Project file**`app.project.file`**Description**

Objet `ExtendScript File` du fichier contenant le projet actuellement ouvert.

**Type**

Objet `File` ou valeur nulle si le projet n'a pas été enregistré ; lecture seule.

**Méthode Project importFile()**`app.project.importFile(importOptions)`**Description**

Importe le fichier spécifié dans l'objet `ImportOptions` indiqué à l'aide des options spécifiées. Equivaut à la commande Fichier > Importer fichier. Crée et renvoie un nouvel objet `FootageItem` à partir du fichier et l'ajoute au tableau des éléments `Item` du projet.

**Paramètres**

<code>ImportOptions</code>	Objet <code>ImportOptions</code> indiquant le fichier à importer ainsi que les options de cette opération. Voir la section « Objet <code>ImportOptions</code> », page 73.
----------------------------	---

**Renvoi**

Objet `FootageItem`.

**Exemple**

```
app.project.importFile(new ImportOptions(File("sample.psd")))
```

**Méthode Project importFileWithDialog()**`app.project.importFileWithDialog()`**Description**

Affiche une boîte de dialogue Importer fichier. Equivalent de la commande Fichier > Importer > Fichier.

**Renvoi**

Tableau d'objets `Item` créé lors d'une importation ; valeur nulle si l'utilisateur annule la boîte de dialogue.

**Méthode Project importPlaceholder()**

`app.project.importPlaceholder(name, width, height, frameRate, duration)`

**Description**

Crée et renvoie un nouvel objet PlaceholderItem et l'ajoute au tableau des éléments Item du projet. Equivaut à la commande Fichier > Importer > Pseudo-élément.

**Paramètres**

<code>name</code>	Chaîne contenant le nom du pseudo-élément.
<code>width</code>	Largeur du pseudo-élément, en pixels. Nombre entier compris dans la plage [4..30000].
<code>height</code>	Hauteur du pseudo-élément, en pixels. Nombre entier compris dans la plage [4..30000].
<code>frameRate</code>	Cadence du pseudo-élément. Valeur à virgule flottante comprise dans la plage [1.0..99.0].
<code>duration</code>	Durée du pseudo-élément, en secondes. Valeur à virgule flottante comprise dans la plage [0.0..10800.0].

**Renvoi**

Objet PlaceholderItem.

**Méthode Project item()**

`app.project.item(index)`

**Description**

Extrait un objet à la position d'index indiquée.

**Paramètres**

<code>index</code>	Position d'index d'un élément, entier. Le premier élément se situe à l'index 1.
--------------------	---

**Renvoi**

Objet Item.

**Attribut Project items**

`app.project.items`

**Description**

Tous les éléments du projet.

**Type**

Objet ItemCollection ; lecture seule.

**Attribut Project linearBlending**

`app.project.linearBlending`

**Description**

True si la fusion linéaire s'applique à ce projet ; sinon false.

**Type**

Booléen ; lecture/écriture.

**Attribut Project numItems**

app.project.numItems

**Description**

Nombre total d'éléments contenus dans le projet, y compris les dossiers et tous les types de métrage.

**Type**

Entier ; lecture seule.

**Exemple**

```
n = app.project.numItems;
alert("There are " + n + " items in this project.")
```

**Méthode Project reduceProject()**

app.project.reduceProject(*array\_of\_items*)

**Description**

Supprime tous les éléments du projet à l'exception des éléments spécifiés. Equivalent de la commande Fichier > Réduire le projet.

**Paramètres**

array_of_items	Tableau contenant les objets Item à conserver.
----------------	--

**Renvoi**

Entier ; nombre total d'éléments supprimés.

**Exemple**

```
var theItems = new Array();
theItems[theItems.length] = app.project.item(1);
theItems[theItems.length] = app.project.item(3);
app.project.reduceProject(theItems);
```

**Méthode Project removeUnusedFootage()**

app.project.removeUnusedFootage()

**Description**

Supprime tout métrage non utilisé du projet. Equivalent de la commande Fichier > Supprimer le métrage inutilisé.

**Paramètres**

Aucun.

**Renvoi**

Entier ; nombre total d'objets FootageItem supprimés.

**Attribut Project renderQueue**`app.project.renderQueue`**Description**

File d'attente de rendu du projet.

**Type**

Objet RenderQueue ; lecture seule.

**Attribut Project rootFolder**`app.project.rootFolder`**Description**

Dossier racine contenant l'ensemble du projet ; il s'agit d'un dossier virtuel qui contient tous les éléments du panneau Project, mais pas les éléments d'autres dossiers de ce même panneau.

**Type**

Objet FolderItem ; lecture seule.

**Méthode Project save()**`app.project.save()``app.project.save(file)`**Description**

Enregistre le projet. Equivaut à la commande Fichier > Enregistrer ou Fichier > Enregistrer sous. Si le projet n'a jamais été enregistré et qu'aucun fichier n'est spécifié, l'utilisateur doit indiquer l'emplacement et le nom du fichier. Spécifiez un objet File pour enregistrer un projet dans un nouveau fichier sans confirmation de l'utilisateur.

**Paramètres**

file	Facultatif. Objet ExtendScript File du fichier à enregistrer.
------	---

**Renvoi**

Aucun.

**Méthode Project saveWithDialog()**`app.project.saveWithDialog()`**Description**

Affiche la boîte de dialogue Enregistrer. L'utilisateur peut indiquer un fichier avec un emplacement pour enregistrer le projet, ou cliquer sur Annuler pour fermer la boîte de dialogue.

**Paramètres**

Aucun.

**Renvoi**

Booléen ; true si le projet a été enregistré.

**Attribut Project selection**

app.project.selection

**Description**

Tous les éléments sélectionnés dans le panneau Projet, triés comme dans ce panneau.

**Type**

Tableau d'objets Item ; lecture seule.

**Méthode Project showWindow()**

app.project.showWindow(*doShow*)

**Description**

Affiche ou masque le panneau Projet.

**Paramètres**

doShow	Lorsque ce paramètre est défini sur true, affiche le panneau Projet. Lorsqu'il est défini sur false, masque le panneau Projet.
--------	--

**Renvoi**

Aucun.

**Attribut Project timecodeBaseType**

app.project.timecodeBaseType

**Description**

Option Base du code temporel, telle que définie dans la boîte de dialogue Paramètres du projet.

**Type**

Valeur énumérée TimecodeBaseType ; écriture/lecture. L'une des suivantes :

TimecodeBaseType.AUTO  
TimecodeBaseType.FPS24  
TimecodeBaseType.FPS25  
TimecodeBaseType.FPS30  
TimecodeBaseType.FPS48  
TimecodeBaseType.FPS50  
TimecodeBaseType.FPS60  
TimecodeBaseType.FPS100

**Attribut Project timecodeDisplayType**

app.project.timecodeDisplayType

**Description**

Mode d'affichage du code temporel, tel que défini dans la boîte de dialogue Paramètres du projet.

**Type**

Valeur énumérée TimecodeDisplayType ; lecture/écriture. L'une des suivantes :

TimecodeDisplayType.TIMECODE

TimecodeDisplayType.FRAMES

TimecodeDisplayType.FEET\_AND\_FRAMES

**Attribut Project timecodeFilmType**

app.project.timecodeFilmType

**Description**

Type de film, tel que défini dans l'option Pieds + images de la boîte de dialogue Paramètres du projet.

**Type**

Valeur énumérée TimecodeFilmType ; lecture/écriture. L'une des suivantes :

TimecodeFilmType.MM16

TimecodeFilmType.MM35

**Attribut Project timecodeNTSCDropFrame**

app.project.timecodeNTSCDropFrame

**Description**

Mode d'affichage du code temporel pour le métrage à 29,97 i/s, tel que défini sous NTSC dans la boîte de dialogue Paramètres du projet.

**Type**

Booléen, true pour l'option Image compensée, false pour l'option Image non compensée ; lecture/écriture.

**Attribut Project transparencyGridThumbnails**

app.project.transparencyGridThumbnails

**Description**

Lorsque cet attribut est défini sur true, les vignettes utilisent le motif de damier de transparence.

**Type**

Booléen ; lecture/écriture.

## Objet Property

`app.project.item(index).layer(index).propertySpec`

### Description

L'objet Property contient des informations (valeur, image clé et expression) relatives à une propriété AE particulière d'un calque. La propriété AE est une valeur, souvent animable, d'un effet, d'un masque ou d'une transformation dans un calque individuel. Pour des exemples d'accès aux propriétés, reportez-vous aux sections « Objet PropertyBase », page 141 et « Méthode PropertyGroup property() », page 150.

- PropertyGroup est une sous-classe de PropertyBase. Toutes les méthodes et tous les attributs de PropertyBase, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez PropertyGroup. Voir la section « Objet PropertyBase », page 141.

*REMARQUE : les objets JavaScript, appelés couramment « propriétés », sont appelés « attributs » dans ce guide pour éviter toute confusion avec la définition du mot « propriété » dans After Effects.*

### Attributs

Attribut	Référence	Description
propertyValueType	« Attribut Property propertyValueType », page 132	Type de valeur stocké dans cette propriété.
value	« Attribut Property value », page 139	Valeur actuelle de la propriété.
hasMin	« Attribut Property hasMin », page 125	Lorsque cet attribut est défini sur true, il existe une valeur minimale autorisée.
hasMax	« Attribut Property hasMax », page 124	Lorsque cet attribut est défini sur true, il existe une valeur maximale autorisée.
minValue	« Attribut Property minValue », page 131	Valeur minimale autorisée.
maxValue	« Attribut Property maxValue », page 131	Valeur maximale autorisée.
isSpatial	« Attribut Property isSpatial », page 125	Lorsque cet attribut est défini sur true, la propriété définit une valeur spatiale.
canVaryOverTime	« Attribut Property canVaryOverTime », page 123	Lorsque cet attribut est défini sur true, la propriété peut être appliquée à une image clé.
isTimeVarying	« Attribut Property isTimeVarying », page 125	Lorsque cet attribut est défini sur true, la propriété dispose d'images clés ou d'une expression activée faisant varier ses valeurs.
numKeys	« Attribut Property numKeys », page 132	Nombre d'images clés sur cette propriété.
unitsText	« Attribut Property unitsText », page 139	Texte décrivant les unités dans lesquelles la valeur est exprimée.
expression	« Attribut Property expression », page 124	Chaîne d'expression pour cette propriété.
canSetExpression	« Attribut Property canSetExpression », page 123	Lorsque cet attribut est défini sur true, l'expression peut être définie par un script.
expressionEnabled	« Attribut Property expressionEnabled », page 124	Lorsque cet attribut est défini sur true, l'expression est utilisée pour générer des valeurs pour la propriété.
expressionError	« Attribut Property expressionError », page 124	Erreur qui s'est éventuellement produite lors de l'évaluation de la dernière expression.

Attribut	Référence	Description
selectedKeys	« Attribut Property selectedKeys », page 133	Toutes les images clés de la propriété.
propertyIndex	« Attribut Property propertyIndex », page 132	Index de position de cette propriété.

### Méthodes

Méthode	Référence	Description
valueAtTime()	« Méthode Property valueAtTime() », page 140	Prend la valeur de la propriété évaluée à un instant donné.
setValue()	« Méthode Property setValue() », page 138	Définit la valeur statique de la propriété.
setValueAtTime()	« Méthode Property setValueAtTime() », page 138	Crée une image clé pour la propriété.
setValuesAtTimes()	« Méthode Property setValuesAtTimes() », page 139	Crée un ensemble d'images clés pour la propriété.
setValueAtKey()	« Méthode Property setValueAtKey() », page 138	Recherche une image clé et attribue la valeur de la propriété à celle-ci.
nearestKeyIndex()	« Méthode Property nearestKeyIndex() », page 131	Prend l'image clé la plus proche du moment spécifié.
keyTime()	« Méthode Property keyTime() », page 130	Prend la valeur du moment où la condition est remplie.
keyValue()	« Méthode Property keyValue() », page 130	Prend la valeur d'une image clé au moment où la condition est remplie.
addKey()	« Méthode Property addKey() », page 123	Ajoute une nouvelle image clé à la propriété à un moment donné.
removeKey()	« Méthode Property removeKey() », page 133	Supprime une image clé de la propriété.
isInterpolationTypeValid()	« Méthode Property isInterpolationTypeValid() », page 125	Lorsque cette méthode est définie sur true, la propriété peut être interpolée.
setInterpolationTypeAtKey()	« Méthode Property setInterpolationTypeAtKey() », page 134	Définit le type d'interpolation pour une clé.
keyInInterpolationType()	« Méthode Property keyInInterpolationType() », page 126	Attribue le type d'interpolation « in » à une clé.
keyOutInterpolationType()	« Méthode Property keyOutInterpolationType() », page 127	Attribue le type d'interpolation « out » à une clé.
setSpatialTangentsAtKey()	« Méthode Property setSpatialTangentsAtKey() », page 136	Définit les vecteurs de tangente « in » et « out » pour une clé.
keyInSpatialTangent()	« Méthode Property keyInSpatialTangent() », page 126	Attribue la tangente spatiale « in » à une clé.
keyOutSpatialTangent()	« Méthode Property keyOutSpatialTangent() », page 127	Attribue la tangente spatiale « out » à une clé.
setTemporalEaseAtKey()	« Méthode Property setTemporalEaseAtKey() », page 137	Définit le lissage temporel « in » et « out » pour une clé.
keyInTemporalEase()	« Méthode Property keyInTemporalEase() », page 126	Attribue le lissage temporel « in » à une clé.

Méthode	Référence	Description
keyOutTemporalEase()	« Méthode Property keyOutTemporalEase() », page 128	Attribue le lissage temporel « out » à une clé.
setTemporalContinuousAtKey()	« Méthode Property setTemporalContinuousAtKey() », page 137	Définit si l'image clé est dotée d'une continuité temporelle.
keyTemporalContinuous()	« Méthode Property keyTemporalContinuous() », page 130	Indique si l'image clé est dotée d'une continuité temporelle.
setTemporalAutoBezierAtKey()	« Méthode Property setTemporalAutoBezierAtKey() », page 136	Définit si l'image clé est dotée du mode Bézier automatique temporel.
keyTemporalAutoBezier()	« Méthode Property keyTemporalAutoBezier() », page 129	Indique si l'image clé est dotée du mode Bézier automatique temporel.
setSpatialContinuousAtKey()	« Méthode Property setSpatialContinuousAtKey() », page 135	Définit si l'image clé est dotée d'une continuité spatiale.
keySpatialContinuous()	« Méthode Property keySpatialContinuous() », page 129	Indique si l'image clé est dotée d'une continuité spatiale.
setSpatialAutoBezierAtKey	« Méthode Property setSpatialAutoBezierAtKey() », page 135	Définit si l'image clé est dotée du mode Bézier automatique spatial.
keySpatialAutoBezier()	« Méthode Property keySpatialAutoBezier() », page 129	Indique si l'image clé est dotée du mode Bézier automatique spatial.
setRovingAtKey()	« Méthode Property setRovingAtKey() », page 134	Définit si l'image clé est itinérante.
keyRoving()	« Méthode Property keyRoving() », page 128	Indique si l'image clé est itinérante.
setSelectedAtKey()	« Méthode Property setSelectedAtKey() », page 135	Spécifie si une image clé est sélectionnée.
keySelected()	« Méthode Property keySelected() », page 128	Indique si une image clé est sélectionnée.

**Exemple : obtention et définition de la valeur d'opacité**

```
var myProperty = myLayer.opacity;
//opacity has propertyValueType of OneD, and is stored as a float
myProperty.setValue(0.5);
// Variable myOpacity is a float value
var myOpacity = myProperty.value;
```

**Exemple : obtention et définition de la valeur d'une position**

```
var myProperty = myLayer.position;
//position has propertyValueType of ThreeD_SPATIAL, and is stored as an array of 3 floats
myProperty.setValue([10.0, 30.0, 0.0] );
// Variable myPosition is an array of 3 floats
var myPosition = myProperty.value;
```

**Exemple : redéfinition de la valeur d'une forme de masque de fermée à ouverte**

```
var myMask = mylayer.mask(1);
var myProperty = myMask.maskPath;
myShape = myProperty.value;
myShape.closed = false;
myProperty.setValue(myShape);
```

**Exemple : obtention de la valeur d'une couleur à un instant donné**

Une couleur est stockée en tant que tableau de quatre valeurs à virgule flottante (r, v, b, opacity). Le script suivant définit la valeur de la composante rouge (red) d'une couleur de lumière à l'instant 4 à la moitié de celle de l'instant 2 :

```
var myProperty = myLight.color;
var colorValue = myProperty.valueAtTime(2,true);
colorValue[0] = 0.5 * colorValue[0];
myProperty.setValueAtTime(4,colorValue);
```

**Exemple : procédure de vérification qu'une échelle calculée par une expression à l'instant 3,5 est la valeur attendue de [10,50]**

```
var myProperty = myLayer.scale;
// false value of preExpression means evaluate the expression
var scaleValue = myProperty.valueAtTime(3.5,false);
if (scaleValue[0] == 10 && scaleValue[1] == 50) {
  alert("hurray");
}
else {
  alert("oops");
}
```

**Exemple : application d'une image clé à une rotation de 0 à 90 et retour**

L'animation dure 10 secondes, et l'image clé centrale se situe au repère 5 secondes. Les propriétés de rotation sont stockées sous la forme d'une valeur OneD.

```
myProperty = myLayer.rotation;
myProperty.setValueAtTime(0, 0);
myProperty.setValueAtTime(5, 90);
myProperty.setValueAtTime(10, 0);
```

**Exemple : modification des valeurs des images clés pour les trois premières images clés d'un texte source**

```
myProperty = myTextLayer.sourceText;
if (myProperty.numKeys < 3) {
  alert("error, I thought there were 3 keyframes");
}
else {
  myProperty.setValueAtKey(1, new TextDocument("key number 1"));
  myProperty.setValueAtKey(2, new TextDocument("key number 2"));
  myProperty.setValueAtKey(3, new TextDocument("key number 3"));
}
```

**Exemple : définition des valeurs à l'aide de la syntaxe de commodité pour la position, la mise à l'échelle, la couleur ou le texte source**

```
// These two are equivalent. The second fills in a default of 0.
myLayer.position.setValue([20, 30, 0]);
myLayer.position.setValue([20, 30]);
// These two are equivalent. The second fills in a default of 100.
myLayer.scale.setValue([50, 50, 100]);
myLayer.scale.setValue([50, 50]);
// These two are equivalent. The second fills in a default of 1.0
```

```
myLight.color.setValue([.8, .3, .1, 1,0]);
myLight.color.setValue([.8, .3, .1]);
// These two are equivalent. The second creates a TextDocument
myTextLayer.sourceText.setValue(new TextDocument("foo"));
myTextLayer.sourceText.setValue("foo");
```

### Méthode Property addKey()

```
app.project.item(index).layer(index).propertySpec.addKey(time)
```

#### Description

Ajoute une nouvelle image clé ou un nouveau repère à la propriété nommée à l'instant donné et renvoie l'index de la nouvelle image clé.

#### Paramètres

time	Moment, en secondes, où ajouter l'image clé. Valeur à virgule flottante. Le début de la composition est 0.
------	--

#### Renvoi

Entier ; index de la nouvelle image clé ou du nouveau repère.

### Attribut Property canSetExpression

```
app.project.item(index).layer(index).propertySpec.canSetExpression
```

#### Description

Lorsque cet attribut est défini sur true, le type de la propriété nommée est celui dont l'expression peut être définie par un script. Voir aussi la section « Attribut Property expression », page 124.

#### Type

Booléen ; lecture seule.

### Attribut Property canVaryOverTime

```
app.project.item(index).layer(index).propertySpec.canVaryOverTime
```

#### Description

Lorsque cet attribut est défini sur true, la propriété nommée peut varier en fonction du temps. C'est-à-dire que les valeurs d'image clé ou d'expression peuvent être écrites dans cette propriété.

#### Type

Booléen ; lecture seule.

**Attribut Property expression**

```
app.project.item(index).layer(index).propertySpec.expression
```

**Description**

Expression pour la propriété nommée. En écriture seule si la valeur `canSetExpression` de la propriété nommée est `true`. La chaîne est évaluée si vous spécifiez une valeur pour cet attribut.

- Si la chaîne contient une expression valide, la valeur de `expressionEnabled` devient `true`.
- Si la chaîne ne contient pas d'expression valide, une erreur est générée et la valeur attribuée à `expressionEnabled` est `false`.
- Si la chaîne de l'attribut est vide, la valeur de `expressionEnable` est `false`, mais aucune erreur n'est générée.

**Type**

Chaîne ; en lecture/écriture si la valeur `canSetExpression` de la propriété nommée est `true`.

**Attribut Property expressionEnabled**

```
app.project.item(index).layer(index).propertySpec.expressionEnabled
```

**Description**

Lorsque cet attribut est défini sur `true`, la propriété nommée utilise l'expression associée pour générer une valeur. Si la valeur est `false`, les informations de l'image clé ou la valeur statique de la propriété sont utilisées. La valeur de cet attribut peut être `true` uniquement si la valeur `true` est attribuée à l'expression `canSetExpression` de la propriété nommée et que l'expression contient une chaîne d'expression valide.

**Type**

Booléen ; lecture/écriture.

**Attribut Property expressionError**

```
app.project.item(index).layer(index).propertySpec.expressionError
```

**Description**

Contient l'erreur éventuellement générée par l'évaluation de la chaîne la plus récemment définie dans l'expression. Si aucune chaîne d'expression n'a été spécifiée ou si la dernière chaîne d'expression a été évaluée sans erreur, l'attribut contient la chaîne vide (`""`).

**Type**

Chaîne ; lecture seule.

**Attribut Property hasMax**

```
app.project.item(index).layer(index).propertySpec.hasMax
```

**Description**

Lorsque cet attribut est défini sur `true`, il existe une valeur maximale autorisée pour la propriété nommée ; sinon, la valeur est `false`.

**Type**

Booléen ; lecture seule.

**Attribut Property hasMin**

`app.project.item(index).layer(index).propertySpec.hasMin`

**Description**

Lorsque cet attribut est défini sur `true`, il existe une valeur minimale autorisée pour la propriété nommée ; sinon, la valeur est `false`.

**Type**

Booléen ; lecture seule.

**Méthode Property isInterpolationTypeValid()**

`app.project.item(index).layer(index).propertySpec.isInterpolationTypeValid(type)`

**Description**

Renvoie `true` si la propriété nommée peut être interpolée à l'aide du type d'interpolation d'image clé spécifié.

**Paramètres**

<code>type</code>	Valeur énumérée <code>KeyframeInterpolationType</code> ; l'une des suivantes : <code>KeyframeInterpolationType.LINEAR</code> <code>KeyframeInterpolationType.BEZIER</code> <code>KeyframeInterpolationType.HOLD</code>
-------------------	---

**Renvoi**

Booléen.

**Attribut Property isSpatial**

`app.project.item(index).layer(index).propertySpec.isSpatial`

**Description**

Lorsque cet attribut est défini sur `true`, la propriété nommée définit une valeur spatiale. Il s'agit par exemple des réglages de position et de point d'effet.

**Type**

Booléen ; lecture seule.

**Attribut Property isTimeVarying**

`app.project.item(index).layer(index).propertySpec.isTimeVarying`

**Description**

Lorsque cet attribut est défini sur `true`, la propriété nommée varie dans le temps (c'est-à-dire qu'elle comporte des images clés ou une expression activée). Si la valeur de `isTimeVarying` est `true`, celle de l'attribut `canVaryOverTime` doit également être `true`.

**Type**

Booléen ; lecture seule.

**Méthode Property keyInInterpolationType()**

```
app.project.item(index).layer(index).propertySpec.keyInInterpolationType(keyIndex)
```

**Description**

Renvoie le type d'interpolation « in » pour l'image clé spécifiée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
----------	--

**Renvoi**

Valeur énumérée KeyframeInterpolationType ; l'une des suivantes :

KeyframeInterpolationType.LINEAR

KeyframeInterpolationType.BEZIER

KeyframeInterpolationType.HOLD

**Méthode Property keyInSpatialTangent()**

```
app.project.item(index).layer(index).propertySpec.keyInSpatialTangent(keyIndex)
```

**Description**

Renvoie la tangente spatiale d'approche pour l'image clé spécifiée, si la propriété nommée est spatiale (c'est-à-dire si la valeur est du type TwoD\_SPATIAL ou ThreeD\_SPATIAL).

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
----------	--

**Renvoi**

Tableau de valeurs à virgule flottante :

- Si la valeur de la propriété est de type PropertyValue.Type.TwoD\_SPATIAL, le tableau contient 2 valeurs à virgule flottante.
- Si la valeur de la propriété est de type PropertyValue.Type.ThreeD\_SPATIAL, le tableau contient 3 valeurs à virgule flottante.
- Si la valeur de la propriété n'est d'aucun de ces deux types, une exception est générée.

**Méthode Property keyInTemporalEase()**

```
app.project.item(index).layer(index).propertySpec.keyInTemporalEase(keyIndex)
```

**Description**

Renvoie le lissage temporel d'approche pour l'image clé spécifiée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
----------	--

**Renvoi**

Tableau d'objets `KeyframeEase` :

- Si la valeur de la propriété est de type `PropertyValue.Type.TwoD`, le tableau contient 2 objets.
- Si la valeur de la propriété est de type `PropertyValue.Type.ThreeD`, le tableau contient 3 objets.
- Pour tout autre type de valeur, le tableau contient 1 objet.

**Méthode Property `keyOutInterpolationType()`**

`app.project.item(index).layer(index).propertySpec.keyOutInterpolationType(keyIndex)`

**Description**

Renvoie le type d'interpolation d'éloignement pour l'image clé spécifiée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
----------	---

**Renvoi**

Valeur énumérée `KeyframeInterpolationType` ; l'une des suivantes :

`KeyframeInterpolationType.LINEAR`

`KeyframeInterpolationType.BEZIER`

`KeyframeInterpolationType.HOLD`

**Méthode Property `keyOutSpatialTangent()`**

`app.project.item(index).layer(index).propertySpec.keyOutSpatialTangent(keyIndex)`

**Description**

Renvoie la tangente spatiale d'éloignement pour l'image clé spécifiée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
----------	---

**Renvoi**

Tableau de valeurs à virgule flottante :

- Si la valeur de la propriété est de type `PropertyValue.Type.TwoD_SPATIAL`, le tableau contient 2 valeurs à virgule flottante.
- Si la valeur de la propriété est de type `PropertyValue.Type.ThreeD_SPATIAL`, le tableau contient 3 valeurs à virgule flottante.
- Si la valeur de la propriété n'est d'aucun de ces deux types, une exception est générée.

**Méthode Property keyOutTemporalEase()**

```
app.project.item(index).layer(index).propertySpec.keyOutTemporalEase(keyIndex)
```

**Description**

Revoie le lissage temporel d'éloignement pour l'image clé spécifiée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
----------	--

**Renvoi**

Tableau d'objets KeyframeEase :

- Si la valeur de la propriété est de type PropertyValue.Type.TwoD, le tableau contient 2 objets.
- Si la valeur de la propriété est de type PropertyValue.Type.ThreeD, le tableau contient 3 objets.
- Pour tout autre type de valeur, le tableau contient 1 objet.

**Méthode Property keyRoving()**

```
app.project.item(index).layer(index).propertySpec.keyRoving(keyIndex)
```

**Description**

Revoie la valeur true si l'image clé est itinérante. La première et la dernière image clé d'une propriété ne peuvent pas être itinérantes ; si vous essayez de les rendre itinérantes, l'opération n'est pas prise en compte et keyRoving() renvoie toujours la valeur false.

Si le type de valeur de la propriété n'est ni TwoD\_SPATIAL ni ThreeD\_SPATIAL, une exception est générée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
----------	--

**Renvoi**

Booléen.

**Méthode Property keySelected()**

```
app.project.item(index).layer(index).propertySpec.keySelected(keyIndex)
```

**Description**

Revoie la valeur true si l'image clé spécifiée est sélectionnée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
----------	--

**Renvoi**

Booléen.

**Méthode Property keySpatialAutoBezier()**

```
app.project.item(index).layer(index).propertySpec.keySpatialAutoBezier(keyIndex)
```

**Description**

Renvoie la valeur true si l'image clé spécifiée comporte une interpolation Bézier automatique spatiale. (Ce type d'interpolation affecte cette image clé uniquement si la valeur true est aussi attribuée à `keySpatialContinuous(keyIndex)`.)

Si le type de valeur de la propriété n'est ni `TwoD_SPATIAL` ni `ThreeD_SPATIAL`, une exception est générée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
----------	---

**Renvoi**

Booléen.

**Méthode Property keySpatialContinuous()**

```
app.project.item(index).layer(index).propertySpec.keySpatialContinuous(keyIndex)
```

**Description**

Renvoie la valeur true si l'image clé spécifiée est dotée d'une continuité spatiale.

Si le type de valeur de la propriété n'est ni `TwoD_SPATIAL` ni `ThreeD_SPATIAL`, une exception est générée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
----------	---

**Renvoi**

Booléen.

**Méthode Property keyTemporalAutoBezier()**

```
app.project.item(index).layer(index).propertySpec.keyTemporalAutoBezier(keyIndex)
```

**Description**

Renvoie la valeur true si l'image clé spécifiée est dotée d'une interpolation Bézier automatique temporelle.

L'interpolation Bézier automatique temporelle affecte cette image clé uniquement si le type d'interpolation d'image clé est `KeyframeInterpolationType.BEZIER` à la fois pour `keyInInterpolationType(keyIndex)` et pour `keyOutInterpolationType(keyIndex)`.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
----------	---

**Renvoi**

Booléen.

**Méthode Property keyTemporalContinuous()**

```
app.project.item(index).layer(index).propertySpec.keyTemporalContinuous(keyIndex)
```

**Description**

Renvoie la valeur true si l'image clé spécifiée est dotée d'une continuité temporelle.

La continuité temporelle affecte cette image clé uniquement si le type d'interpolation d'image clé est `KeyframeInterpolationType.BEZIER` à la fois pour `keyInInterpolationType(keyIndex)` et pour `keyOutInterpolationType(keyIndex)`.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
----------	---

**Renvoi**

Booléen.

**Méthode Property keyTime()**

```
app.project.item(index).layer(index).propertySpec.keyTime(keyIndex)
app.project.item(index).layer(index).propertySpec.keyTime(markerComment)
```

**Description**

Recherche le repère ou l'image clé spécifié(e) et renvoie le moment auquel il/elle apparaît.

Si aucune image clé ou aucun repère ne correspond à l'argument, cette méthode génère une exception et une erreur s'affiche.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
markerComment	Chaîne de commentaire associée à un repère (voir la section « Attribut MarkerValue comment », page 99).

**Renvoi**

Valeur à virgule flottante.

**Méthode Property keyValue()**

```
app.project.item(index).layer(index).propertySpec.keyValue(keyIndex)
app.project.item(index).layer(index).propertySpec.keyValue(markerComment)
```

**Description**

Recherche le repère ou l'image clé spécifié(e) et renvoie sa valeur actuelle.

Si aucune image clé ou aucun repère ne correspond à l'argument, cette méthode génère une exception et une erreur s'affiche.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
markerComment	Chaîne de commentaire associée à un repère (voir la section « Attribut MarkerValue comment », page 99).

**Renvoi**

Valeur à virgule flottante.

**Attribut Property maxValue**

`app.project.item(index).layer(index).propertySpec.maxValue`

**Description**

Valeur maximale autorisée pour la propriété nommée. Si l'attribut `hasMax` est `false`, une exception se produit et une erreur est générée.

**Type**

Valeur à virgule flottante ; lecture seule.

**Attribut Property minValue**

`app.project.item(index).layer(index).propertySpec.minValue`

**Description**

Valeur minimale autorisée pour la propriété nommée. Si l'attribut `hasMin` est `false`, une exception se produit et une erreur est générée.

**Type**

Valeur à virgule flottante ; lecture seule.

**Méthode Property nearestKeyIndex()**

`app.project.item(index).layer(index).propertySpec.nearestKeyIndex(time)`

**Description**

Renvoie l'index de l'image clé la plus proche de l'instant spécifié.

**Paramètres**

time	Temps, en secondes ; valeur à virgule flottante. Le début de la composition est 0.
------	--

**Renvoi**

Entier.

**Attribut Property numKeys**

```
app.project.item(index).layer(index).propertySpec.numKeys
```

**Description**

Nombre d'images clés dans la propriété nommée. Si la valeur est 0, la propriété ne contient pas d'images clés.

**Type**

Entier ; lecture seule.

**Attribut Property propertyIndex**

```
app.project.item(index).layer(index).propertySpec.propertyIndex
```

**Description**

Index de position de la propriété nommée. La première propriété se trouve à la position d'index 1.

**Type**

Entier ; lecture seule.

**Attribut Property propertyValueType**

```
app.project.item(index).layer(index).propertySpec.propertyValueType
```

**Description**

Type de valeur stocké dans la propriété nommée. L'énumération `PropertyValueType` possède une valeur pour chaque type de données pouvant être stockées dans une propriété et en être extraites. Chaque type de données est stocké et extrait dans une structure de type différent. Tous les objets de propriété stockent les données dans l'une de ces catégories.

Par exemple, une propriété spatiale 3D (telle qu'une position de calque) est stockée en tant que tableau de trois valeurs à virgule flottante. Lors de la définition d'une valeur pour la position, utilisez un tableau de ce type :

```
mylayer.property("position").setValue([10,20,0]);
```

En revanche, une propriété de forme (telle que la forme de masque d'un calque) est stockée en tant qu'objet `Shape`. Lors de la définition d'une valeur pour une forme, utilisez un objet `Shape`, comme suit :

```
var myShape = new Shape();
myShape.vertices = [[0,0],[0 100],[100 100],[100,0]];
var myMask = mylayer.property("ADBE Mask Parade").property(1);
myMask.property("ADBE Mask Shape").setValue(myShape);
```

**Type**

Valeur énumérée `PropertyValueType` ; lecture/écriture. L'une des suivantes :

<code>PropertyValueType.NO_VALUE</code>	Ne stocke aucune donnée.
<code>PropertyValueType.ThreeD_SPATIAL</code>	Tableau de trois valeurs à virgule flottante positionnelles. Exemple de valeur d'un point d'ancrage : [10.0, 20.2, 0.0].
<code>PropertyValueType.ThreeD</code>	Tableau de trois valeurs à virgule flottante quantitatives. Exemple de valeur d'échelle : [100.0, 20.2, 0.0].

PropertyValueType.TwoD_SPATIAL	Tableau de deux valeurs à virgule flottante positionnelles. Exemple de valeur d'un point d'ancrage : [5.1, 10.0].
PropertyValueType.TwoD	Tableau de deux valeurs à virgule flottante quantitatives. Exemple de valeur d'échelle : [5.1, 100.0].
PropertyValueType.OneD	Valeur à virgule flottante.
PropertyValueType.COLOR	Tableau de quatre valeurs à virgule flottante, comprises dans la plage [0.0..1.0]. Par exemple : [0.8, 0.3, 0.1, 1.0]
PropertyValueType.CUSTOM_VALUE	Type non disponible ; vous ne pouvez ni obtenir ni définir de valeurs pour les propriétés avec ce type.
PropertyValueType.MARKER	Objet MarkerValue ; voir la section « Objet MarkerValue », page 98.
PropertyValueType.LAYER_INDEX	Entier ; la valeur 0 signifie aucun calque.
PropertyValueType.MASK_INDEX	Entier ; la valeur 0 signifie aucun masque.
PropertyValueType.SHAPE	Objet Shape ; voir la section « Objet Shape », page 166.
PropertyValueType.TEXT_DOCUMENT	Objet TextDocument ; voir la section « Objet TextDocument », page 173.

### Méthode Property removeKey()

```
app.project.item(index).layer(index).propertySpec.removeKey(keyIndex)
```

#### Description

Supprime l'image clé spécifiée de la propriété nommée. S'il n'existe aucune image clé avec l'index indiqué, cela génère une exception et une erreur s'affiche.

Si une image clé est supprimée, les numéros d'index restants changent. Pour supprimer plusieurs images clés, vous devez commencer par le numéro d'index le plus élevé afin de vous assurer que les index restants référencent toujours les mêmes images clés après chaque suppression.

#### Paramètres

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
----------	--

#### Renvoi

Aucun.

### Attribut Property selectedKeys

```
app.project.item(index).layer(index).propertySpec.selectedKeys
```

#### Description

Index de toutes les images clés sélectionnées dans la propriété nommée. Si aucune image clé n'est sélectionnée ou si la propriété ne comporte aucune image clé, un tableau vide est renvoyé.

#### Type

Tableau d'entiers ; lecture seule.

**Méthode Property setInterpolationTypeAtKey()**

`app.project.item(index).layer(index).propertySpec.setInterpolationTypeAtKey(keyIndex, inType, outType)`

**Description**

Définit les types d'interpolation « in » et « out » pour l'image clé spécifiée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
inType	Type d'interpolation d'approche. Valeur énumérée <code>KeyframeInterpolationType</code> ; l'une des suivantes :  <code>KeyframeInterpolationType.LINEAR</code> <code>KeyframeInterpolationType.BEZIER</code> <code>KeyframeInterpolationType.HOLD</code>
outType	(Facultatif) Type d'interpolation d'éloignement. Si rien n'est indiqué, la valeur attribuée à <code>inType</code> est « out ». Valeur énumérée <code>KeyframeInterpolationType</code> ; l'une des suivantes :  <code>KeyframeInterpolationType.LINEAR</code> <code>KeyframeInterpolationType.BEZIER</code> <code>KeyframeInterpolationType.HOLD</code>

**Renvoi**

Aucun.

**Méthode Property setRovingAtKey()**

`app.project.item(index).layer(index).propertySpec.setRovingAtKey(keyIndex, newVal)`

**Description**

Active ou désactive le mode itinérant pour l'image clé spécifiée. La première et la dernière image clé d'une propriété ne peuvent pas être itinérantes ; si vous essayez de les rendre itinérantes, l'opération n'est pas prise en compte et `keyRoving()` renvoie toujours la valeur `false`.

Si le type de valeur de la propriété n'est ni `TwoD_SPATIAL` ni `ThreeD_SPATIAL`, une exception est générée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
newVal	<code>True</code> pour activer le mode itinérant, <code>false</code> pour le désactiver.

**Renvoi**

Aucun.

**Méthode Property setSelectedAtKey()**

```
app.project.item(index).layer(index).propertySpec.setSelectedAtKey(keyIndex, onOff)
```

**Description**

Sélectionne ou désélectionne l'image clé spécifiée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
onOff	True pour sélectionner l'image clé, false pour la désélectionner.

**Renvoi**

Aucun.

**Méthode Property setSpatialAutoBezierAtKey()**

```
app.project.item(index).layer(index).propertySpec.setSpatialAutoBezierAtKey(keyIndex, newVal)
```

**Description**

Active ou désactive l'interpolation Bézier automatique spatiale pour l'image clé spécifiée.

Si le type de valeur de la propriété n'est ni TwoD\_SPATIAL ni ThreeD\_SPATIAL, une exception est générée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
newVal	True pour activer l'interpolation Bézier automatique spatiale, false pour la désactiver.

**Renvoi**

Aucun.

**Méthode Property setSpatialContinuousAtKey()**

```
app.project.item(index).layer(index).propertySpec.setSpatialContinuousAtKey(keyIndex, newVal)
```

**Description**

Active ou désactive la continuité spatiale pour l'image clé spécifiée.

Si le type de valeur de la propriété n'est ni TwoD\_SPATIAL ni ThreeD\_SPATIAL, une exception est générée.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
newVal	True pour activer la continuité spatiale, false pour la désactiver.

**Renvoi**

Aucun.

**Méthode Property setSpatialTangentsAtKey()**

`app.project.item(index).layer(index).propertySpec.setSpatialTangentsAtKey(keyIndex, inTangent, outTangent)`

**Description**

Définit les vecteurs de tangente d'approche et d'éloignement pour l'image clé spécifiée.

Si le type de valeur de la propriété n'est ni `TwoD_SPATIAL` ni `ThreeD_SPATIAL`, une exception est générée.

**Paramètres**

<code>keyIndex</code>	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
<code>inTangent</code>	Vecteur de tangente d'approche. Tableau de 2 ou 3 valeurs à virgule flottante. <ul style="list-style-type: none"> <li>• Si la valeur de la propriété est de type <code>PropertyValue.Type.TwoD_SPATIAL</code>, le tableau contient 2 valeurs.</li> <li>• Si la valeur de la propriété est de type <code>PropertyValue.Type.ThreeD_SPATIAL</code>, le tableau contient 3 valeurs.</li> </ul>
<code>outTangent</code>	(Facultatif) Vecteur de tangente d'éloignement. Si rien n'est indiqué, la valeur attribuée à <code>inTangent</code> est « out ». Tableau de 2 ou 3 valeurs à virgule flottante. <ul style="list-style-type: none"> <li>• Si la valeur de la propriété est de type <code>PropertyValue.Type.TwoD_SPATIAL</code>, le tableau contient 2 valeurs.</li> <li>• Si la valeur de la propriété est de type <code>PropertyValue.Type.ThreeD_SPATIAL</code>, le tableau contient 3 valeurs.</li> </ul>

**Renvoi**

Aucun.

**Méthode Property setTemporalAutoBezierAtKey()**

`app.project.item(index).layer(index).propertySpec.setTemporalAutoBezierAtKey(keyIndex, newVal)`

**Description**

Active ou désactive l'interpolation Bézier automatique temporelle pour l'image clé spécifiée. Lorsque cette option est activée, elle affecte cette image clé uniquement si la valeur `true` est également attribuée à `keySpatialContinuous(keyIndex)`.

**Paramètres**

<code>keyIndex</code>	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
<code>newVal</code>	<code>True</code> pour activer l'interpolation Bézier automatique temporelle, <code>false</code> pour la désactiver.

**Renvoi**

Aucun.

**Méthode Property setTemporalContinuousAtKey()**

```
app.project.item(index).layer(index).propertySpec.setTemporalContinuousAtKey(keyIndex, newVal)
```

**Description**

Active ou désactive la continuité temporelle pour l'image clé spécifiée.

Si la continuité temporelle est activée, elle affecte cette image clé uniquement si le type d'interpolation d'image clé est `KeyframeInterpolationType.BEZIER` à la fois pour `keyInInterpolationType(keyIndex)` et pour `keyOutInterpolationType(keyIndex)`.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
newVal	True pour activer la continuité temporelle, false pour la désactiver.

**Renvoi**

Aucun.

**Méthode Property setTemporalEaseAtKey()**

```
app.project.item(index).layer(index).propertySpec.setTemporalEaseAtKey(keyIndex, inTemporalEase, outTemporalEase)
```

**Description**

Définit le lissage temporel d'approche et d'éloignement pour l'image clé spécifiée. Voir la section « `Objet KeyframeEase` », page 81.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode <code>addKey</code> ou <code>nearestKeyIndex</code> .
inTemporalEase	Lissage temporel d'approche. Tableau contenant 1, 2 ou 3 objets <code>KeyframeEase</code> . <ul style="list-style-type: none"> <li>• Si la valeur de la propriété est de type <code>PropertyValueType.TwoD</code>, le tableau contient 2 objets.</li> <li>• Si la valeur de la propriété est de type <code>PropertyValueType.ThreeD</code>, le tableau contient 3 objets.</li> <li>• Pour tous les autres types de valeur, le tableau contient 1 objet.</li> </ul>
outTemporalEase	(Facultatif) Lissage temporel d'éloignement. Si rien n'est indiqué, la valeur attribuée à <code>inTemporalEase</code> est celle du lissage d'éloignement. Tableau contenant 1, 2 ou 3 objets <code>KeyframeEase</code> . <ul style="list-style-type: none"> <li>• Si la valeur de la propriété est de type <code>PropertyValueType.TwoD</code>, le tableau contient 2 objets.</li> <li>• Si la valeur de la propriété est de type <code>PropertyValueType.ThreeD</code>, le tableau contient 3 objets.</li> <li>• Pour tous les autres types de valeur, le tableau contient 1 objet.</li> </ul>

**Renvoi**

Aucun.

**Méthode Property setValue()**

```
app.project.item(index).layer(index).propertySpec.setValue(newValue)
```

**Description**

Définit la valeur statique d'une propriété n'ayant pas d'image clé.

Si la propriété nommée comporte des images clés, cette méthode génère une exception et une erreur est affichée. Pour définir la valeur d'une propriété comportant des images clés, reportez-vous à la section « Méthode Property setValueAtTime() », page 138 ou « Méthode Property setValueAtKey() », page 138.

**Paramètres**

newValue	Valeur appropriée pour le type de propriété défini ; voir la section « Attribut Property propertyValueType », page 132.
----------	---

**Renvoi**

Aucun.

**Méthode Property setValueAtKey()**

```
app.project.item(index).layer(index).propertySpec.setValueAtKey(keyIndex, newValue)
```

**Description**

Recherche l'image clé spécifiée et définit sa valeur.

Si la propriété nommée ne comporte aucune image clé ou aucune image clé ayant l'index spécifié, cette méthode génère une exception et une erreur s'affiche.

**Paramètres**

keyIndex	Index de l'image clé. Entier compris dans la plage [1..numKeys], tel que renvoyé par la méthode addKey ou nearestKeyIndex.
newValue	Valeur appropriée pour le type de propriété défini ; voir la section « Attribut Property propertyValueType », page 132.

**Renvoi**

Aucun.

**Méthode Property setValueAtTime()**

```
app.project.item(index).layer(index).propertySpec.setValueAtTime(time, newValue)
```

**Description**

Définit la valeur d'une image clé au moment indiqué. Crée une nouvelle image clé pour la propriété nommée, si elle n'existe pas déjà pour le moment indiqué, et en définit la valeur.

**Paramètres**

time	Temps, en secondes. Valeur à virgule flottante. Le début de la composition est 0.
newValue	Valeur appropriée pour le type de propriété défini ; voir la section « Attribut Property propertyValueType », page 132.

**Renvoi**

Aucun.

### Méthode Property setValuesAtTimes()

`app.project.item(index).layer(index).propertySpec.setValuesAtTimes(times, newValues)`

#### Description

Définit les valeurs d'un ensemble d'images clés à des moments spécifiés. Crée une nouvelle image clé pour la propriété nommée, si elle n'existe pas déjà pour un moment indiqué, et en définit la valeur.

Les moments et les valeurs sont exprimés sous forme de tableaux de longueur identique.

#### Paramètres

<code>times</code>	Tableau de valeurs temporelles, en secondes. Chaque point temporel est une valeur à virgule flottante. Le début de la composition est 0.
<code>newValues</code>	Tableau de valeurs appropriées pour le type de propriété défini ; voir la section « Attribut Property valueType », page 132.

#### Renvoi

Aucun.

### Attribut Property unitsText

`app.project.item(index).layer(index).propertySpec.unitsText`

#### Description

Texte décrivant les unités dans lesquelles la valeur est exprimée.

#### Type

Chaîne ; lecture seule.

### Attribut Property value

`app.project.item(index).layer(index).propertySpec.value`

#### Description

Valeur de la propriété nommée à l'instant courant.

- Lorsque `expressionEnabled` est true, la valeur de l'expression évaluée est renvoyée.
- S'il existe des images clés, la valeur de ces images à l'instant courant est renvoyée.
- Sinon, c'est une valeur statique qui est renvoyée.

Le type de valeur renvoyé dépend du type de valeur de la propriété. Voir les exemples pour « Objet Property », page 119.

#### Type

Valeur appropriée pour le type de propriété défini (voir la section « Attribut Property valueType », page 132) ; lecture seule.

**Méthode Property valueAtTime()**

```
app.project.item(index).layer(index).propertySpec.valueAtTime(time, preExpression)
```

**Description**

Valeur de la propriété nommée telle qu'elle a été évaluée à l'instant spécifié.

Notez que le type de valeur renvoyé n'est pas explicite ; il peut varier en fonction de la propriété évaluée.

**Paramètres**

time	Temps, en secondes ; valeur à virgule flottante. Le début de la composition est 0.
preExpression	Si la propriété est dotée d'une expression dont la valeur est true, la valeur pour l'instant spécifié est renvoyée sans que l'expression lui soit appliquée. Si ce paramètre est false, le résultat de l'évaluation de l'expression à l'instant spécifié est renvoyé.  Ignoré si la propriété ne dispose pas d'une expression associée.

**Renvoi**

Valeur appropriée pour le type de propriété défini (voir la section « Attribut Property propertyValueType », page 132).

## Objet PropertyBase

`app.project.item(index).layer(index).propertySpec`

### Description

L'accès aux propriétés des calques se fait à l'aide de leur nom, en utilisant diverses syntaxes d'expression (contrôlées par les préférences de l'application). Voici par exemple toutes les façons d'accéder aux propriétés du groupe Effects :

```
var effect1 = app.project.item(1).layer(1).effect("Add Grain")("Viewing Mode");
var effect1again = app.project.item(1).layer(1).effect.addGrain.viewingMode;
var effect1againtoo = app.project.item(1).layer(1)("Effects").addGrain.viewingMode;
var effect1againtoo2 = app.project.item(1).layer(1)("Effects")("Add Grain")("Viewing Mode");
```

Voir aussi la section « Méthode PropertyGroup property() », page 150.

- PropertyBase est à la fois la classe de base de Property et de PropertyGroup ; par conséquent, les méthodes et attributs de PropertyBase sont disponibles lorsque vous travaillez avec les propriétés et les groupes de propriétés. Voir les sections « Objet Property », page 119 et « Objet PropertyGroup », page 148.

### Références non valides

S'il se produit une action qui modifie un objet de façon assez significative pour en fausser la référence, les références à cet objet dans le script risquent de générer des erreurs. Dans les cas simples, ce problème est facile à résoudre. Par exemple, si vous supprimez un objet, toute référence à l'objet effacé génère un message d'avertissement de type « Objet non valide ».

```
var layer1 = app.project.item(1).layer(1);
layer1.remove();
alert(layer1.name); // invalid reference to deleted object
```

Si vous faites référence à une propriété AE dans un objet supprimé, vous obtenez le même avertissement :

```
var layer1 = app.project.item(1).layer(1);
var layer1position = layer1.transform.position;
layer1.remove();
alert(layer1position.value); // invalid reference to property in selected object
```

Le problème est plus difficile à résoudre si la propriété a été supprimée d'un groupe de propriétés. After Effects génère l'erreur « Objet non valide » lorsque vous référencez cet objet ou d'autres éléments du groupe, car les positions d'index ont changé. Par exemple :

```
var effect1 = app.project.item(1).layer(1).effect(1);
var effect2 = app.project.item(1).layer(1).effect(2);
var effect2param = app.project.item(1).layer(1).effect(2).blendWithOriginal;
effect1.remove();
alert(effect2.name); // invalid reference because group index positions have changed
```

### Attributs

Attribut	Référence	Description
name	« Attribut PropertyBase name », page 145	Nom de la propriété.

Attribut	Référence	Description
matchName	« Attribut PropertyBase matchName », page 145	Nom spécial de la propriété utilisé pour créer des noms de chemin d'accès uniques.
propertyIndex	« Attribut PropertyBase propertyIndex », page 146	Index de la propriété dans son groupe parent.
propertyDepth	« Attribut PropertyBase propertyDepth », page 146	Nombre de niveaux de groupes parent entre cette propriété et le calque la contenant.
propertyType	« Attribut PropertyBase propertyType », page 147	Type de propriété.
parentProperty	« Attribut PropertyBase parentProperty », page 146	Groupe parent immédiat de cette propriété.
isModified	« Attribut PropertyBase isModified », page 144	Lorsque cet attribut est true, la propriété a été modifiée depuis sa création.
canSetEnabled	« Attribut PropertyBase canSetEnabled », page 143	Lorsque cet attribut est true, l'interface utilisateur affiche une icône en forme d'œil pour cette propriété.
enabled	« Attribut PropertyBase enabled », page 144	Lorsque cet attribut est true, la propriété est activée.
active	« Attribut PropertyBase active », page 142	Lorsque cet attribut est true, la propriété est active.
elided	« Attribut PropertyBase elided », page 143	Lorsque cet attribut est true, cette propriété ne s'affiche pas dans l'interface utilisateur.
isEffect	« Attribut PropertyBase isEffect », page 144	Lorsque cet attribut est true, la propriété est un effet.
isMask	« Attribut PropertyBase isMask », page 144	Lorsque cet attribut est true, la propriété est un masque.
selected	« Attribut PropertyBase selected », page 147	Lorsque cet attribut est true, la propriété est sélectionnée.

### Méthodes

Méthode	Référence	Description
propertyGroup()	« Méthode PropertyBase propertyGroup() », page 146	Extrait le groupe parent pour cette propriété.
remove()	« Méthode PropertyBase remove() », page 147	Supprime cet élément du projet.
moveTo()	« Méthode PropertyBase moveTo() », page 145	Déplace cette propriété à un nouvel endroit dans son groupe parent.
duplicate()	« Méthode PropertyBase duplicate() », page 143	Duplique cet objet de propriété.

### Attribut PropertyBase active

`app.project.item(index).layer(index).propertySpec.active`

#### Description

Lorsque cet attribut est true, la propriété est active. Pour un calque, cela correspond au paramètre de l'icône en forme d'œil. Pour un effet et toutes les propriétés, il s'agit de l'équivalent de l'attribut `enabled`.

#### Type

Booléen ; en lecture/écriture si la valeur de `canSetEnabled` est true ; en lecture seule si la valeur de `canSetEnabled` est false.

**Attribut PropertyBase canSetEnabled**

```
app.project.item(index).layer(index).propertySpec.canSetEnabled
```

**Description**

Si la valeur est true, vous pouvez affecter la valeur d'attribut enabled. En règle générale, cet attribut est true si l'interface utilisateur affiche une icône en forme d'œil pour cette propriété ; la valeur true s'applique à tous les calques.

**Type**

Booléen ; lecture seule.

**Méthode PropertyBase duplicate()**

```
app.project.item(index).layer(index).propertySpec.duplicate()
```

**Description**

Si cette propriété est un enfant d'un groupe indexé, un nouvel objet PropertyBase ayant les mêmes valeurs d'attribut que celle-ci est créé et renvoyé.

Si cette propriété n'est pas un enfant de groupe indexé, la méthode génère une exception et un message d'erreur s'affiche.

Un groupe indexé est de type PropertyType.INDEXED\_GROUP ; voir la section « Attribut PropertyBase propertyType », page 147.

**Paramètres**

Aucun.

**Renvoi**

Objet PropertyBase.

**Attribut PropertyBase elided**

```
app.project.item(index).layer(index).propertySpec.elided
```

**Description**

Lorsque cet attribut est true, la propriété est un groupe utilisé pour organiser d'autres propriétés. La propriété ne s'affiche pas dans l'interface utilisateur et ses propriétés enfant ne sont pas indentées dans le panneau Montage.

Par exemple : dans le cas d'un calque de texte comportant deux animations et aucune propriété indentée, vous voyez :

```
Text
Path Options
More Options
Animator 1
Animator 2
```

Dans cet exemple, « Animator 1 » et « Animator 2 » sont contenus dans un PropertyBase appelé « Text Animators ». Ce groupe parent ne s'affiche pas dans l'interface utilisateur ; par conséquent, les deux propriétés enfant ne sont pas indentées dans le panneau Montage.

**Type**

Booléen ; lecture seule.

**Attribut PropertyBase enabled**

```
app.project.item(index).layer(index).propertySpec.enabled
```

**Description**

Lorsque cet attribut est true, la propriété est activée. Cet attribut correspond au paramètre de l'icône en forme d'œil, si celle-ci est présente ; sinon, la valeur par défaut est true.

**Type**

Booléen ; en lecture/écriture si la valeur de `canSetEnabled` est true ; en lecture seule si la valeur de `canSetEnabled` est false.

**Attribut PropertyBase isEffect**

```
app.project.item(index).layer(index).propertySpec.isEffect
```

**Description**

Lorsque cet attribut est true, la propriété est un PropertyGroup d'effets.

**Type**

Booléen ; lecture seule.

**Attribut PropertyBase isMask**

```
app.project.item(index).layer(index).propertySpec.isMask
```

**Description**

Lorsque cet attribut est true, la propriété est un PropertyGroup de masques.

**Type**

Booléen ; lecture seule.

**Attribut PropertyBase isModified**

```
app.project.item(index).layer(index).propertySpec.isModified
```

**Description**

Lorsque cet attribut est true, la propriété a été modifiée depuis sa création.

**Type**

Booléen ; lecture seule.

### Attribut PropertyBase matchName

`app.project.item(index).layer(index).propertySpec.matchName`

#### Description

Nom spécial de la propriété utilisé pour créer des noms de chemin d'accès uniques. Le nom interne ne s'affiche pas, mais vous pouvez y faire référence dans les scripts. Chaque propriété possède un identificateur de nom interne unique. Les noms internes sont stables d'une version à l'autre indépendamment du nom qui s'affiche (valeur de l'attribut `name`) ou des modifications apportées à l'application. Contrairement au nom d'affichage, il n'est pas traduit.

Un groupe indexé peut ne pas avoir de valeur `name` ; en revanche, il dispose toujours d'une valeur `matchName`. (Un groupe indexé est de type `PropertyType.INDEXED_GROUP` ; voir la section « Attribut PropertyBase propertyType », page 147.)

#### Type

Chaîne ; lecture seule.

### Méthode PropertyBase moveTo()

`app.project.item(index).layer(index).propertySpec.moveTo(newIndex)`

#### Description

Déplace cette propriété à un nouvel endroit dans son groupe de propriétés parent.

Cette méthode n'est valide que pour les enfants des groupes indexés ; si ce n'est pas le cas ou si la valeur d'index n'est pas valide, une exception est générée et une erreur s'affiche. (Un groupe indexé est de type `PropertyType.INDEXED_GROUP` ; voir la section « Attribut PropertyBase propertyType », page 147.)

#### Paramètres

<code>newIndex</code>	Nouvelle position d'index dans le groupe où sera placée cette propriété. Nombre entier.
-----------------------	---

#### Renvoi

Aucun.

### Attribut PropertyBase name

`app.project.item(index).layer(index).propertySpec.name`

#### Description

Nom d'affichage de la propriété. (Comparer avec « Attribut PropertyBase matchName », page 145.)

Ne définissez pas la valeur `name` si la propriété n'est pas un enfant d'un groupe indexé (c'est-à-dire, un groupe de propriétés du type `PropertyType.INDEXED_GROUP` ; voir la section « Attribut PropertyBase propertyType », page 147).

#### Type

Chaîne ; en lecture/écriture pour un enfant d'un groupe indexé, sinon en lecture seule.

**Attribut PropertyBase parentProperty**

```
app.project.item(index).layer(index).propertySpec.parentProperty
```

**Description**

Groupe de propriétés se trouvant être le parent immédiat de cette propriété, ou valeur nulle si PropertyBase est un calque.

**Type**

Objet PropertyGroup ou valeur nulle ; lecture seule.

**Attribut PropertyBase propertyDepth**

```
app.project.item(index).layer(index).propertySpec.propertyDepth
```

**Description**

Nombre de niveaux de groupes parent entre cette propriété et le calque la contenant. Valeur 0 pour un calque.

**Type**

Entier ; lecture seule.

**Méthode PropertyBase propertyGroup()**

```
app.project.item(index).layer(index).propertySpec.propertyGroup()
app.project.item(index).layer(index).propertySpec.propertyGroup(countUp)
```

**Description**

Extrait l'objet PropertyGroup pour un groupe ancêtre de cette propriété au niveau spécifié de la hiérarchie parent-enfant.

**Paramètres**

countUp	Facultatif. Nombre de niveaux à inclure dans la hiérarchie parent-enfant. Nombre entier compris dans la plage [1..propertyDepth]. La valeur par défaut de 1 permet d'accéder au parent immédiat.
---------	--

**Renvoi**

Objet PropertyGroup, ou valeur nulle si le décompte atteint le calque contenant.

**Attribut PropertyBase propertyIndex**

```
app.project.item(index).layer(index).propertySpec.propertyIndex
```

**Description**

Index de position de cette propriété à l'intérieur de son groupe parent, s'il s'agit d'un enfant d'un groupe indexé (groupe de propriétés du type PropertyType.INDEXED\_GROUP ; voir la section « Attribut PropertyBase propertyType », page 147).

**Type**

Entier ; lecture seule.

**Attribut PropertyBase propertyType**

```
app.project.item(index).layer(index).propertySpec.propertyType
```

**Description**

Type de cette propriété.

**Type**

Valeur énumérée PropertyType ; lecture/écriture. L'une des suivantes :

PropertyType.PROPERTY	Propriété unique telle que la position ou le zoom.
PropertyType.INDEXED_GROUP	Groupe de propriétés dont les membres disposent d'un index et d'un nom qu'il est possible de modifier. Les effets et les masques sont des groupes indexés. Par exemple, la propriété <code>masks</code> d'un calque renvoie à un nombre variable de masques différents en fonction du numéro d'index.
PropertyType.NAMED_GROUP	Groupe de propriétés dont il est impossible de modifier les noms des membres. Les calques sont des groupes nommés.

**Méthode PropertyBase remove()**

```
app.project.item(index).layer(index).propertySpec.remove()
```

**Description**

Supprime cette propriété de son groupe parent. S'il s'agit d'un groupe de propriétés, les propriétés enfant sont également supprimées.

Cette méthode n'est valide que pour les enfants des groupes indexés ; si ce n'est pas le cas ou si la valeur d'index n'est pas valide, une exception est générée et une erreur s'affiche. (Un groupe indexé est de type PropertyType.INDEXED\_GROUP ; voir la section « Attribut PropertyBase propertyType », page 147.)

Cette méthode peut être appelée sur une propriété d'animation de texte (c'est-à-dire toute animation ayant été affectée à un calque de texte).

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Attribut PropertyBase selected**

```
app.project.item(index).layer(index).propertySpec.selected
```

**Description**

Lorsque cet attribut est true, la propriété est sélectionnée. Définissez la valeur true pour sélectionner la propriété, ou false pour la désélectionner.

L'échantillonnage répété de cet attribut pour un grand nombre de propriétés peut ralentir les performances du système. Pour lire la totalité des propriétés sélectionnées d'une composition ou d'un calque, utilisez l'attribut `selectedProperties` d'un objet `Comp` ou `Layer`.

**Type**

Booléen ; en lecture/écriture pour un groupe de propriétés de masque ou d'effet, sinon en lecture seule.

## Objet PropertyGroup

`app.project.item(index).layer(index).propertyGroupSpec`

### Description

L'objet PropertyGroup représente un groupe de propriétés. Il peut contenir des objets Property et d'autres objets PropertyGroup. Les groupes Property peuvent être imbriqués pour former une hiérarchie parent-enfant, partant d'un objet Layer au niveau supérieur (racine) et descendant jusqu'à un objet Property unique, tel que le contour progressif du troisième masque. Pour traverser la hiérarchie du groupe, utilisez les méthodes et les attributs PropertyBase ; voir la section « Méthode PropertyBase propertyGroup() », page 146.

Pour obtenir des exemples sur la manière d'accéder aux propriétés et aux groupes de propriétés, reportez-vous à la section « Objet PropertyBase », page 141.

- PropertyGroup est une sous-classe de PropertyBase. Toutes les méthodes et tous les attributs de PropertyBase, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez PropertyGroup. Voir la section « Objet PropertyBase », page 141.
- PropertyGroup est une classe de base de MaskPropertyGroup. Les méthodes et les attributs de PropertyGroup sont disponibles lorsque vous travaillez avec des groupes de masques. Voir la section « Objet MaskPropertyGroup », page 102.

### Attributs

Attribut	Référence	Description
numProperties	« Attribut PropertyGroup numProperties », page 149	Nombre de propriétés indexées dans le groupe.

### Méthodes

Méthode	Référence	Description
property()	« Méthode PropertyGroup property() », page 150	Extrait un groupe ou une propriété d'un membre.
canAddProperty()	« Méthode PropertyGroup canAddProperty() », page 149	Indique si une propriété peut être ajoutée au groupe.
addProperty()	« Méthode PropertyGroup addProperty() », page 148	Ajoute une propriété au groupe.

### Méthode PropertyGroup addProperty()

`app.project.item(index).layer(index).propertyGroupSpec.addProperty(name)`

### Description

Crée et renvoie un objet PropertyBase, avec le nom spécifié, et l'ajoute à ce groupe.

En général, vous pouvez uniquement ajouter des propriétés à un groupe indexé, c'est-à-dire du type PropertyType.INDEXED\_GROUP ; voir la section « Attribut PropertyBase propertyType », page 147. La seule exception est la propriété d'animation de texte, qu'il est possible d'ajouter à un groupe nommé (groupe de propriétés du type PropertyType.NAMED\_GROUP).

Si cette méthode ne peut pas créer de propriété avec le nom spécifié, une exception est générée. Pour vérifier que vous pouvez ajouter une propriété donnée à ce groupe, appelez canAddProperty avant d'utiliser cette méthode. (Voir la section « Méthode PropertyGroup canAddProperty() », page 149.)

**Paramètres**

name	<p>Nom d'affichage ou nom interne de la propriété à ajouter. (Voir la section « Attribut PropertyBase matchName », page 145.)</p> <p>Les noms suivants sont pris en charge :</p> <ul style="list-style-type: none"> <li>• Tout nom interne d'une propriété pouvant être ajoutée via l'interface utilisateur. Par exemple : « ADBE Mask Atom », « ADBE Paint Atom », « ADBE Text Position », « ADBE Text Anchor Point ».</li> <li>• Lors d'un ajout à ADBE Mask Parade : « ADBE Mask Atom », « Mask ».</li> <li>• Lors d'un ajout à ADBE Effect Parade : tout effet par nom interne, tel que « ADBE Bulge », « ADBE Glo2 », « APC Vegas ».</li> <li>• Tout effet par nom d'affichage, tel que « Bulge », « Glow », « Vegas ».</li> <li>• Pour les animations de texte, « ADBE Text Animator ».</li> <li>• En ce qui concerne les sélecteurs : le sélecteur de plage a pour nom « ADBE Text Selector », le sélecteur de tremblement a pour nom « ADBE Text Wiggly Selector » et le sélecteur d'expressions a pour nom « ADBE Text Expressible Selector ».</li> </ul>
------	--

**Renvoi**

Objet PropertyBase.

**Méthode PropertyGroup canAddProperty()**

```
app.project.item(index).layer(index).propertyGroupSpec.canAddProperty(name)
```

**Description**

Renvoie true si une propriété portant le nom donné peut être ajoutée à ce groupe de propriétés. Par exemple : vous pouvez uniquement ajouter un masque à un groupe de masques. Les seuls arguments d'entrée autorisés sont « Mask » et « ADBE Mask Atom ».

```
maskGroup.canAddProperty("mask"); //returns true
maskGroup.canAddProperty("ADBE Mask Atom"); //returns true
maskGroup.canAddProperty("blend"); // returns false
```

**Paramètres**

name	Nom d'affichage ou nom interne de la propriété à vérifier. (Voir la section « Méthode PropertyGroup addProperty() », page 148.)
------	---

**Renvoi**

Booléen.

**Attribut PropertyGroup numProperties**

```
app.project.item(index).layer(index).propertyGroupSpec.numProperties
```

**Description**

Nombre de propriétés indexées dans ce groupe.

Pour les calques, cette méthode renvoie une valeur de 3, correspondant aux groupes de masques, d'effets et de suivi de trajectoire, qui sont les groupes indexés dans le calque. Toutefois, les calques comportent également de nombreuses autres propriétés uniquement accessibles par nom ; voir la section « Méthode PropertyGroup property() », page 150.

**Type**

Entier ; lecture seule.

**Méthode PropertyGroup property()**

```
app.project.item(index).layer(index).propertyGroupSpec.property(index)
```

```
app.project.item(index).layer(index).propertyGroupSpec.property(name)
```

**Description**

Cherche et renvoie une propriété enfant de ce groupe, spécifiée soit par son index, soit par son nom.

Pour spécifier un nom, vous pouvez utiliser la même syntaxe que celle des expressions. Toutes les expressions ci-dessous sont autorisées et sont équivalentes :

```
mylayer.position
```

```
mylayer("position")
```

```
mylayer.property("position")
```

```
mylayer(1)
```

```
mylayer.property(1)
```

Certaines des propriétés d'un calque, telles que la position et le zoom, sont accessibles uniquement par le nom.

Lorsque vous utilisez le nom pour rechercher une propriété se trouvant à plusieurs niveaux inférieurs, vous devez appeler cette méthode plusieurs fois. Par exemple, l'appel suivant effectue une recherche deux niveaux plus bas et renvoie le premier masque du groupe de masques :

```
myLayer.property("ADBE Masks").property(1)
```

**Paramètres**

index	L'index de la propriété enfant est, dans ce cas, un groupe indexé. Nombre entier compris dans la plage [0..numProperties].
name	Nom de la propriété enfant. Il peut s'agir de : <ul style="list-style-type: none"> <li>• Tout nom interne.</li> <li>• Tout nom utilisé dans la syntaxe d'expression « parenthesis style » (c'est-à-dire le nom d'affichage ou le nom anglais compact).</li> <li>• Tout nom utilisé dans la syntaxe d'expression « intercap style ».</li> </ul> Pour connaître les noms de propriété pris en charge, consultez le tableau ci-dessous.

**Renvoi**

Objet PropertyBase ou valeur nulle si aucune propriété enfant avec le nom de chaîne donné n'a été trouvée.

**Propriétés accessibles par nom**

A partir d'un calque quelconque	<ul style="list-style-type: none"> <li>• "ADBE Mask Parade" ou "Masks"</li> <li>• "ADBE Effect Parade" ou "Effects"</li> <li>• "ADBE MTrackers" ou "Motion Trackers"</li> </ul>
---------------------------------	---

A partir d'un AVLayer	<ul style="list-style-type: none"> <li>• "Anchor Point" ou "anchorPoint"</li> <li>• "Position" ou "position"</li> <li>• "Scale" ou "scale"</li> <li>• "Rotation" ou "rotation"</li> <li>• "Z Rotation" ou "zRotation" ou "Rotation Z" ou "rotationZ"</li> <li>• "Opacity" ou "opacity"</li> <li>• "Marker" ou "marker"</li> </ul>
A partir d'un AVLayer avec une source non fixe	<ul style="list-style-type: none"> <li>• "Time Remap" ou "timeRemapEnabled"</li> </ul>
A partir d'un AVLayer avec un composant audio	<ul style="list-style-type: none"> <li>• "Audio Levels" ou "audioLevels"</li> </ul>
A partir d'un calque de caméra	<ul style="list-style-type: none"> <li>• "Zoom" ou "zoom"</li> <li>• "Depth of Field" ou "depthOfField"</li> <li>• "Focus Distance" ou "focusDistance"</li> <li>• "Aperture" ou "aperture"</li> <li>• "Blur Level" ou "blurLevel"</li> </ul>
A partir d'un calque de lumière	<ul style="list-style-type: none"> <li>• "Intensity" ou "intensity"</li> <li>• "Color" ou "color"</li> <li>• "Cone Angle" ou "coneAngle"</li> <li>• "Cone Feather" ou "coneFeather"</li> <li>• "Shadow Darkness" ou "shadowDarkness"</li> <li>• "Shadow Diffusion" ou "shadowDiffusion"</li> <li>• "Casts Shadows" ou "castsShadows"</li> </ul>
A partir d'un calque 3D	<ul style="list-style-type: none"> <li>• "Accepts Shadows" ou "acceptsShadows"</li> <li>• "Accepts Lights" ou "acceptsLights"</li> <li>• "Ambient" ou "ambient"</li> <li>• "Diffuse" ou "diffuse"</li> <li>• "Specular" ou "specular"</li> <li>• "Shininess" ou "shininess"</li> <li>• "Casts Shadows" ou "castsShadows"</li> <li>• "Light Transmission" ou "lightTransmission"</li> <li>• "Metal" ou "metal"</li> </ul>
A partir d'un calque de caméra ou de lumière ou d'un calque 3D	<ul style="list-style-type: none"> <li>• "X Rotation" ou "xRotation" ou "Rotation X" ou "rotationX"</li> <li>• "Y Rotation" ou "yRotation" ou "Rotation Y" ou "rotationY"</li> <li>• "Orientation" ou "orientation"</li> </ul>
A partir d'un calque de texte	<ul style="list-style-type: none"> <li>• "Source Text" ou "sourceText" ou "Text" ou "text"</li> </ul>
A partir d'un PropertyGroup "ADBE Mask Parade"	<ul style="list-style-type: none"> <li>• "ADBE Mask Atom"</li> </ul>
A partir d'un PropertyGroup "ADBE Mask Atom"	<ul style="list-style-type: none"> <li>• "ADBE Mask Shape" ou "maskShape" ou "maskPath"</li> <li>• "ADBE Mask Feather" ou "maskFeather"</li> <li>• "ADBE Mask Opacity" ou "maskOpacity"</li> <li>• "ADBE Mask Offset" ou "maskOffset"</li> </ul>

**Exemples**

- 1** Si un calque nommé « myLayer » comporte un effet Flou de la bordure (Box Blur), vous pouvez extraire l'effet de l'une des façons suivantes :

```
myLayer.property("Effects").property("Box Blur");  
myLayer.property("Effects").property("boxBlur");  
myLayer.property("Effects").property("ADBE Box Blur");
```

- 2** Si un calque nommé « myLayer » comporte un masque nommé « Mask 1 », vous pouvez l'extraire comme suit :

```
myLayer.property("Masks").property("Mask 1");
```

- 3** Pour obtenir la valeur du centre (Bulge Center) d'un effet de renflement (Bulge), vous pouvez procéder de l'une des façons suivantes :

```
myLayer.property("Effects").property("Bulge").property("Bulge Center");  
myLayer.property("Effects").property("Bulge").property("bulgeCenter");
```

## Objet RenderQueue

app.project.renderQueue

### Description

L'objet RenderQueue représente le processus d'automatisation du rendu, les données et les fonctions disponibles via le panneau File d'attente de rendu d'un projet After Effects donné. Les attributs donnent accès aux éléments contenus dans la file d'attente de rendu et à leur état de rendu. Les méthodes peuvent démarrer, suspendre et arrêter le processus de rendu.

L'objet RenderQueueItem donne accès aux paramètres spécifiques d'un élément dont le rendu doit être effectué. Voir la section « Objet RenderQueueItem », page 156.

### Attributs

Attribut	Référence	Description
rendering	« Attribut RenderQueue rendering », page 155	Lorsque cet attribut est true, un rendu est en cours.
numItems	« Attribut RenderQueue numItems », page 154	Nombre total d'éléments se trouvant dans la file d'attente de rendu.
items	« Attribut RenderQueue items », page 154	Collection des éléments se trouvant dans la file d'attente de rendu.

### Méthodes

Méthode	Référence	Description
showWindow()	« Méthode RenderQueue showWindow() », page 155	Affiche ou masque le panneau File d'attente de rendu.
render()	« Méthode RenderQueue render() », page 154	Démarre le processus de rendu ; ne renvoie rien tant que le rendu n'est pas terminé.
pauseRendering()	« Méthode RenderQueue pauseRendering() », page 154	Suspend ou poursuit le processus de rendu.
stopRendering()	« Méthode RenderQueue stopRendering() », page 155	Arrête le processus de rendu.
item()	« Méthode RenderQueue item() », page 153	Extrait de la collection un élément de la file d'attente de rendu.

### Méthode RenderQueue item()

app.project.renderQueue.item(*index*)

### Description

Extrait un élément donné de la collection d'éléments Item.

### Paramètres

index	Index de position de l'élément. Nombre entier compris dans la plage [0..numItems].
-------	--

### Renvoi

Objet RenderQueueItem.

**Attribut RenderQueue items**`app.project.renderQueue.items`**Description**

Collection de tous les éléments se trouvant dans la file d'attente de rendu. Voir la section « [Objet RenderQueueItem](#) », page 156.

**Type**

Objet `RQItemCollection` ; lecture/écriture.

**Attribut RenderQueue numItems**`app.project.renderQueue.numItems`**Description**

Nombre total d'éléments se trouvant dans la file d'attente de rendu.

**Type**

Entier ; lecture seule.

**Méthode RenderQueue pauseRendering()**`app.project.renderQueue.pauseRendering(pause)`**Description**

Suspend le processus de rendu en cours, ou reprend un processus de rendu ayant été suspendu. Cela revient à cliquer sur Pause dans le panneau File d'attente de rendu pendant un rendu. Vous pouvez lancer cette méthode à partir d'un rappel `onStatusChanged` ou `onError`. Voir les sections « [Attribut RenderQueueItem onStatusChanged](#) », page 158 et « [Attribut Application onError](#) », page 26.

**Paramètres**

<code>pause</code>	True pour suspendre le processus de rendu en cours, false pour reprendre un processus de rendu ayant été suspendu.
--------------------	--

**Renvoi**

Aucun.

**Méthode RenderQueue render()**`app.project.renderQueue.render()`**Description**

Démarre le processus de rendu. Cela revient à cliquer sur Rendu dans le panneau File d'attente de rendu. Cette méthode ne renvoie rien tant que le rendu n'est pas terminé. Pour suspendre ou arrêter le processus de rendu, appelez `pauseRendering()` ou `stopRendering()` à partir d'un rappel `onError` ou `onStatusChanged`.

- Pour réagir aux erreurs au cours d'un processus de rendu, définissez une fonction de rappel dans `app.onError` ; voir la section « [Attribut Application onError](#) », page 26.
- Pour réagir aux modifications d'état d'un élément donné lorsque le processus de rendu est en cours, définissez une fonction de rappel dans `RenderQueueItem.onStatusChanged`, dans l'objet `RenderQueueItem` associé ; voir la section « [Attribut RenderQueueItem onStatusChanged](#) », page 158.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Attribut RenderQueue rendering**

`app.project.renderQueue.rendering`

**Description**

Lorsque cet attribut est true, le processus de rendu est en cours ou momentanément suspendu. Lorsque cet attribut est false, le processus est arrêté.

**Type**

Booléen ; lecture seule.

**Méthode RenderQueue showWindow()**

`app.project.renderQueue.showWindow(doShow)`

**Description**

Affiche ou masque le panneau File d'attente de rendu.

**Paramètres**

<code>doShow</code>	Lorsque ce paramètre est true, le panneau File d'attente de rendu est affiché. Lorsque ce paramètre est false, le panneau est masqué.
---------------------	---

**Renvoi**

Aucun.

**Méthode RenderQueue stopRendering()**

`app.project.renderQueue.stopRendering()`

**Description**

Arrête le processus de rendu. Cela revient à cliquer sur Arrêter dans le panneau File d'attente de rendu au cours d'un rendu. Vous pouvez lancer cette méthode à partir d'un rappel `onStatusChanged` ou `onError`. Voir les sections « Attribut RenderQueueItem onStatusChanged », page 158 et « Attribut Application onError », page 26.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

## Objet RenderQueueItem

`app.project.renderQueue.items(index)`

### Description

L'objet `RenderQueueItem` représente un élément individuel dans la file d'attente de rendu. Il donne accès aux paramètres spécifiques d'un élément dont le rendu doit être effectué. Créez un objet en ajoutant une composition à la file d'attente de rendu avec l'objet `RQItemCollection` ; voir la section « Méthode `RQItemCollection add()` », page 163.

### Attributs

Attribut	Référence	Description
<code>numOutputModules</code>	« Attribut <code>RenderQueueItem numOutputModules</code> », page 158	Nombre total de modules de sortie affectés à l'élément.
<code>render</code>	« Attribut <code>RenderQueueItem render</code> », page 160	Lorsque cet attribut est true, le rendu de l'élément est déclenché au démarrage de la file d'attente.
<code>startTime</code>	« Attribut <code>RenderQueueItem startTime</code> », page 160	Heure à laquelle le processus de rendu de l'élément a commencé.
<code>elapsedSeconds</code>	« Attribut <code>RenderQueueItem elapsedSeconds</code> », page 158	Temps de rendu écoulé pour cet élément.
<code>timeSpanStart</code>	« Attribut <code>RenderQueueItem timeSpanStart</code> », page 162	Heure de début dans la composition dont le rendu doit être effectué.
<code>timeSpanDuration</code>	« Attribut <code>RenderQueueItem timeSpanDuration</code> », page 161	Durée de la composition dont le rendu doit être effectué.
<code>skipFrames</code>	« Attribut <code>RenderQueueItem skipFrames</code> », page 160	Nombre d'images à ignorer lors du rendu de cet élément.
<code>comp</code>	« Attribut <code>RenderQueueItem comp</code> », page 157	Composition dont le rendu doit être effectué par cet élément.
<code>outputModules</code>	« Attribut <code>RenderQueueItem outputModules</code> », page 159	Collection de modules de sortie pour cet élément.
<code>templates</code>	« Attribut <code>RenderQueueItem templates</code> », page 161	Ensemble de modèles de paramètres de rendu.
<code>status</code>	« Attribut <code>RenderQueueItem status</code> », page 161	Etat de rendu en cours de l'élément.
<code>onStatusChanged</code>	« Attribut <code>RenderQueueItem onStatusChanged</code> », page 158	Fonction de rappel appelée lors d'un processus de rendu si l'état d'un élément est modifié.
<code>logType</code>	« Attribut <code>RenderQueueItem logType</code> », page 158	Type de journal pour cet élément.

### Méthodes

Méthode	Référence	Description
<code>outputModule()</code>	« Méthode <code>RenderQueueItem outputModule()</code> », page 159	Extrait un module de sortie pour l'élément.
<code>remove()</code>	« Méthode <code>RenderQueueItem remove()</code> », page 159	Supprime l'élément de la file d'attente de rendu.
<code>saveAsTemplate()</code>	« Méthode <code>RenderQueueItem saveAsTemplate()</code> », page 160	Enregistre un nouveau modèle de paramètres de rendu.

Méthode	Référence	Description
applyTemplate()	« Méthode RenderQueueItem applyTemplate() », page 157	Applique un modèle de paramètres de rendu.
duplicate	« Méthode RenderQueueItem duplicate() », page 157	Duplique cet élément.

### Méthode RenderQueueItem applyTemplate()

`app.project.renderQueue.item.applyTemplate(templateName)`

#### Description

Applique un modèle de paramètres de rendu à l'élément. Voir aussi « Méthode RenderQueueItem saveAsTemplate() », page 160 et « Attribut RenderQueueItem templates », page 161.

#### Paramètres

templateName	Chaîne contenant le nom du modèle à appliquer.
--------------	--

#### Renvoi

Aucun.

### Attribut RenderQueueItem comp

`app.project.renderQueue.item(index).comp`

#### Description

Composition dont le rendu doit être effectué par cet élément de la file d'attente de rendu. Pour modifier la composition, vous devez supprimer cet élément de la file d'attente de rendu et en créer un nouveau.

#### Type

Objet CompItem ; lecture seule.

### Méthode RenderQueueItem duplicate()

`app.project.renderQueue.item(index).duplicate()`

#### Description

Crée un doublon de cet élément et l'ajoute dans cette file d'attente de rendu.

#### Paramètres

Aucun.

#### Renvoi

Objet RenderQueueItem.

**Attribut RenderQueueItem elapsedSeconds**

`app.project.renderQueue.item(index).elapsedSeconds`

**Description**

Nombre de secondes passées à rendre cet élément.

**Type**

Entier, ou valeur nulle si l'élément n'a pas été rendu ; lecture seule.

**Attribut RenderQueueItem logType**

`app.project.renderQueue.item(index).outputModule.logType`

**Description**

Type de journal pour cet élément, indiquant les événements à enregistrer lorsque l'élément est en cours de rendu.

**Type**

Valeur énumérée `LogType` ; écriture/lecture. L'une des suivantes :

`LogType.ERRORS_ONLY`

`LogType.ERRORS_AND_SETTINGS`

`LogType.ERRORS_AND_PER_FRAME_INFO`

**Attribut RenderQueueItem numOutputModules**

`app.project.renderQueue.item(index).numOutputModules`

**Description**

Nombre total de modules de sortie affectés à cet élément.

**Type**

Entier ; lecture seule.

**Attribut RenderQueueItem onStatusChanged**

`app.project.renderQueue.item(index).onStatusChanged`

**Description**

Nom de la fonction de rappel appelée chaque fois que la valeur de l'attribut `RenderQueueItem.status` change. Voir la section « Attribut `RenderQueueItem.status` », page 161.

Vous ne pouvez pas modifier les éléments de la file d'attente de rendu ni l'application lorsque le processus de rendu est en cours ou qu'il a été suspendu ; vous pouvez toutefois utiliser cette fonction de rappel pour suspendre ou arrêter le processus de rendu. Voir les sections « Méthode `RenderQueue.pauseRendering()` », page 154 et « Méthode `RenderQueue.stopRendering()` », page 155.

Voir aussi la section « Attribut `Application.onError` », page 26.

**Type**

Chaîne de nom de fonction, ou valeur nulle si aucune fonction n'a été attribuée.

**Exemple**

```
function myStatusChanged() {
  alert(app.project.renderQueue.item(1).status)
}
app.project.renderQueue.item(1).onStatusChanged = myStatusChanged();
app.project.renderQueue.item(1).render = false; //changes status and shows dialog
```

**Attribut RenderQueueItem outputModules**

```
app.project.renderQueue.item(index).outputModules
```

**Description**

Collection de modules de sortie pour l'élément.

**Type**

Objet OMCollection ; lecture/écriture.

**Méthode RenderQueueItem outputModule()**

```
app.project.renderQueue.item(index).outputModule(index)
```

**Description**

Extrait un module de sortie avec la position d'index spécifiée.

**Paramètres**

<code>index</code>	Index de position du module de sortie. Nombre entier compris dans la plage [1..numOutputModules].
--------------------	---

**Renvoi**

Objet OutputModule.

**Méthode RenderQueueItem remove()**

```
app.project.renderQueue.item(index).remove()
```

**Description**

Supprime cet élément de la file d'attente de rendu.

**Paramètres**

Aucun.

**Renvoi**

Aucun.

**Attribut RenderQueueItem render**`app.project.renderQueue.item(index).render`**Description**

Lorsque cet attribut est défini sur true, l'élément est rendu au démarrage de la file d'attente. Si la valeur est true, `RenderQueueItem.status` est défini sur `RQItemStatus.QUEUED`. Si la valeur est false, `status` est défini sur `RQItemStatus.UNQUEUED`.

**Type**

Booléen ; lecture/écriture.

**Méthode RenderQueueItem saveAsTemplate()**`app.project.renderQueue.item(index).saveAsTemplate(name)`**Description**

Enregistre les paramètres de rendu actuels des éléments en tant que nouveau modèle avec le nom spécifié.

**Paramètres**

<code>name</code>	Chaîne contenant le nom du nouveau modèle.
-------------------	--

**Renvoi**

Aucun.

**Attribut RenderQueueItem skipFrames**`app.project.renderQueue.item(index).skipFrames`**Description**

Nombre d'images à ignorer lors du rendu de cet élément. Utilisé pour obtenir des tests de rendu plus rapides qu'un rendu complet.

Une valeur égale à 0 signifie qu'aucune image n'est ignorée et produit un rendu régulier de toutes les images. Une valeur égale à 1 signifie qu'une image sur deux est ignorée. Cela équivaut à « rendering on twos ». Plus la valeur est élevée, plus le nombre d'images ignorées est important.

La durée totale reste inchangée. Si, par exemple, `skip` prend la valeur 1, la moitié du nombre d'images sera produite pour une sortie de séquence, alors que, pour une sortie de film, la durée de chaque image sera doublée.

**Type**

Nombre entier compris dans la plage [0..99]. Lecture/écriture.

**Attribut RenderQueueItem startTime**`app.project.renderQueue.item(index).startTime`**Description**

Date et heure auxquelles le rendu de cet élément a commencé.

**Type**

Objet Date, ou valeur nulle si le rendu de l'élément n'a pas commencé ; lecture seule.

**Attribut RenderQueueItem status**

`app.project.renderQueue.item(index).status`

**Description**

Etat du rendu en cours de l'élément.

**Type**

Valeur énumérée `RQItemStatus` ; lecture seule. L'une des suivantes :

<code>RQItemStatus.WILL_CONTINUE</code>	Le processus de rendu a été interrompu.
<code>RQItemStatus.NEEDS_OUTPUT</code>	Absence de chemin de sortie valide pour l'élément.
<code>RQItemStatus.UNQUEUED</code>	L'élément de rendu figure dans le panneau File d'attente de rendu, mais n'est pas prêt pour le rendu.
<code>RQItemStatus.QUEUED</code>	Composition prête pour le rendu.
<code>RQItemStatus.RENDERING</code>	Composition en cours de rendu.
<code>RQItemStatus.USER_STOPPED</code>	Processus de rendu arrêté par l'utilisateur ou un script.
<code>RQItemStatus.ERR_STOPPED</code>	Processus de rendu arrêté suite à une erreur.
<code>RQItemStatus.DONE</code>	Processus de rendu de l'élément terminé.

**Attribut RenderQueueItem templates**

`app.project.renderQueue.item(index).templates`

**Description**

Les noms de tous les modèles de paramètres de rendu sont disponibles pour cet élément. Voir aussi « Méthode `RenderQueueItem saveAsTemplate()` », page 160.

**Type**

Tableau de chaînes ; lecture seule.

**Attribut RenderQueueItem timeSpanDuration**

`app.project.renderQueue.item(index).timeSpanDuration`

**Description**

Durée, en secondes, de la composition à rendre. La durée est déterminée en déduisant l'heure de début de l'heure de fin. Définir cette valeur revient à définir une heure de fin personnalisée dans la boîte de dialogue Paramètres de rendu.

**Type**

Valeur à virgule flottante ; lecture/écriture.

**Attribut RenderQueueItem timeSpanStart**

`app.project.renderQueue.item(index).timeSpanStart`

**Description**

Heure de la composition, en secondes, à laquelle le processus de rendu doit démarrer. Définir cette valeur revient à définir une heure de début personnalisée dans la boîte de dialogue Paramètres de rendu.

**Type**

Valeur à virgule flottante ; lecture/écriture.

## Objet RQItemCollection

`app.project.renderQueue.items`

### Description

RQItemCollection contient tous les éléments de la file d'attente de rendu d'un projet, tels qu'ils apparaissent dans le panneau File d'attente de rendu du projet. La collection permet d'accéder aux objets RenderQueueItem et de les créer à partir de compositions. Le premier objet RenderQueueItem de la collection se trouve à la position d'index 1. Voir la section « Objet RenderQueueItem », page 156.

- RQItemCollection est une sous-classe de Collection. Toutes les méthodes et tous les attributs de Collection sont disponibles lorsque vous utilisez RQItemCollection. Voir la section « Objet Collection », page 51.

### Méthodes

Méthode	Référence	Description
<code>add()</code>	« Méthode RQItemCollection <code>add()</code> », page 163	Ajoute une composition à la file d'attente de rendu.

### Méthode RQItemCollection `add()`

`app.project.renderQueue.items.add(comp)`

### Description

Ajoute une composition à la file d'attente de rendu, ce qui crée un RenderQueueItem.

### Paramètres

<code>comp</code>	Objet Comptem de la composition à ajouter.
-------------------	--

### Renvoi

Objet RenderQueueItem.

## Objet Settings

### Description

L'objet Settings fournit un moyen aisé de gérer les paramètres pour les scripts. Les paramètres sont enregistrés dans le fichier des préférences d'After Effects et sont conservés d'une session de l'application à l'autre. Les paramètres sont identifiés par section et par clé dans le fichier. Chaque nom de clé est associé à une valeur. Dans le fichier de préférences, les noms de section sont indiqués entre crochets et entre guillemets, et les noms de clé entre guillemets sous le nom de section. Toutes les valeurs sont des chaînes.

Vous pouvez créer de nouveaux paramètres avec cet objet et accéder aux paramètres existants.

### Méthodes

Méthode	Référence	Description
<code>saveSetting()</code>	« Méthode Settings <code>saveSetting()</code> », page 165	Enregistre une valeur par défaut pour un paramètre.
<code>getSetting()</code>	« Méthode Settings <code>getSetting()</code> », page 164	Extrait une valeur de paramètre.
<code>haveSetting()</code>	« Méthode Settings <code>haveSetting()</code> », page 164	Indique si le paramètre spécifié a été attribué.

### Méthode Settings `getSetting()`

*app.settings.getSetting(sectionName, keyName)*

### Description

Extrait du fichier de préférences la valeur d'un élément de préférences de création de script.

### Paramètres

<code>sectionName</code>	Chaîne contenant le nom d'une section de paramètres.
<code>keyName</code>	Chaîne contenant le nom de clé d'un élément de paramètre.

### Renvoi

Chaîne.

### Exemple

Si vous avez enregistré un paramètre avec le nom de clé « Aligned Clone » dans la section « Eraser - Paint Settings », vous pouvez en extraire la valeur à l'aide du script suivant :

```
var n = app.settings.getSetting("Eraser - Paint Settings", "Aligned Clone");
alert("The setting is " + n);
```

### Méthode Settings `haveSetting()`

*app.settings.haveSetting(sectionName, keyName)*

### Description

Renvoie true si l'élément de préférences de création de script existe et qu'il est doté d'une valeur.

**Paramètres**

sectionName	Chaîne contenant le nom d'une section de paramètres.
keyName	Chaîne contenant le nom de clé d'un élément de paramètre.

**Renvoi**

Booléen.

**Méthode Settings saveSetting()**

*app.settings.saveSetting(sectionName, keyName, value)*

**Description**

Enregistre une valeur par défaut pour un élément de préférences de création de script.

**Paramètres**

sectionName	Chaîne contenant le nom d'une section de paramètres.
keyName	Chaîne contenant le nom de clé d'un élément de paramètre.
value	Chaîne contenant la nouvelle valeur.

**Renvoi**

Aucun.

## Objet Shape

```
app.project.item(index).layer(index).property(index).property("maskShape").value
```

### Description

L'objet Shape contient des informations décrivant une forme d'un calque de forme ou la forme du contour d'un masque. Ces informations correspondent à la valeur des propriétés AE « Tracé du masque » et de la propriété AE « Tracé » d'un calque de forme. Utilisez le constructeur, `new Shape()`, pour créer un nouvel objet Shape vide, puis définissez chaque attribut pour spécifier la forme.

Une forme est dotée d'un ensemble de points d'ancrage, ou *sommets*, et dispose de deux poignées directrices, ou *vecteurs de tangente*, pour chaque point d'ancrage. Le vecteur de tangente (dans un masque non RotoBézier) détermine la direction de la ligne qui relie deux points d'ancrage. Il existe un vecteur de tangente d'approche et un vecteur de tangente d'éloignement associés à chaque sommet de la forme.

La valeur de la tangente s'exprime sous forme de coordonnées  $x, y$  relatives au sommet associé. Par exemple, une tangente de  $[-1,-1]$  se situe au-dessus et à gauche du sommet et présente une pente de 45 degrés, quelle que soit la position réelle du sommet. Plus une poignée est longue, plus sa portée est grande. Ainsi, un segment de forme d'approche demeure plus près du vecteur de tangente pour une valeur `inTangent` de  $[-2,-2]$  que pour une valeur `inTangent` de  $[-1,-1]$ , bien que les deux s'approchent du sommet en provenant de la même direction.

Si une forme est ouverte, les valeurs `inTangent` du premier sommet et `outTangent` du sommet final sont ignorées. Sinon, ces deux vecteurs doivent spécifier les poignées directrices du segment de connexion final s'éloignant du sommet final et revenant vers le premier sommet.

Les tangentes des masques RotoBézier sont calculées automatiquement. (Voir la section « Attribut `MaskPropertyGroup rotoBezier` », page 104.) Si une forme est utilisée dans un masque RotoBézier, les valeurs de tangente sont ignorées. Cela signifie que, pour les masques RotoBézier, vous pouvez construire une forme en définissant uniquement l'attribut de `sommet` et en attribuant une valeur nulle aux vecteurs `inTangent` et `outTangent`. Lorsque vous accédez à la nouvelle forme, ses valeurs de tangente sont remplies avec des valeurs calculées de façon automatique.

### Exemple : création d'un masque carré

Un carré est une forme fermée constituée de quatre sommets. Par défaut, les valeurs `inTangent` et `outTangent` des segments de droite sont égales à 0 et n'ont pas besoin d'être définies explicitement.

```
var myShape = new Shape();
myShape.vertices = [[0,0], [0 100], [100 100], [100,0]];
myShape.closed = true;
```

### Exemple : création d'un masque en forme de U

Un U est une forme ouverte constituée des quatre mêmes sommets que ceux utilisés dans le carré :

```
var myShape = new Shape();
myShape.vertices = [[0,0], [0 100], [100 100], [100,0]];
myShape.closed = false;
```

### Exemple : création d'un ovale

Un ovale est une forme fermée constituée de 4 sommets et dotée de valeurs `inTangents` et `outTangents` :

```
var myShape = new Shape();
myShape.vertices = [[300,50],[200 150],[300 250],[400 150]];
myShape.inTangents = [[55.23,0],[0,-55.23],[-55.23,0],[0,55.23]];
myShape.outTangents = [[-55.23,0],[0,55.23],[55.23,0],[0,-55.23]];
myShape.closed = true;
```

**Attributs**

Attribut	Référence	Description
closed	« Attribut Shape closed », page 167	Lorsque cet attribut est défini sur true, la forme est une courbe fermée.
vertices	« Attribut Shape vertices », page 168	Points d'ancrage de la forme.
inTangents	« Attribut Shape inTangents », page 167	Vecteurs de tangente s'approchant des sommets de la forme.
outTangents	« Attribut Shape outTangents », page 167	Vecteurs de tangente s'éloignant des sommets de la forme.

**Attribut Shape closed**

*shapeObject.value.closed*

**Description**

Lorsque cet attribut est défini sur true, le premier et le dernier sommets sont reliés pour former une courbe fermée. Lorsque cet attribut est défini sur false, le segment de fermeture n'est pas dessiné.

**Type**

Booléen ; lecture/écriture.

**Attribut Shape inTangents**

*shapeObject.value.inTangents*

**Description**

Vecteurs de tangente, ou poignées directrices, d'approche associés aux sommets de la forme. Définissez chaque vecteur sous forme de tableau de deux valeurs à virgule flottante et rassemblez les vecteurs dans un tableau de longueur identique à celui des sommets.

La valeur par défaut de chaque tangente est [0,0]. Si la forme de masque n'est pas RotoBézier, le segment obtenu est un segment de droite.

Si la forme est un masque RotoBézier, toutes les valeurs sont ignorées et les tangentes sont calculées de façon automatique.

**Type**

Tableau de paires de valeurs à virgule flottante ; lecture/écriture.

**Attribut Shape outTangents**

*shapeObject.value.outTangents*

**Description**

Vecteurs de tangente, ou poignées directrices, d'éloignement associés aux sommets de la forme. Définissez chaque vecteur sous forme de tableau de deux valeurs à virgule flottante et rassemblez les vecteurs dans un tableau de longueur identique à celui des sommets.

La valeur par défaut de chaque tangente est [0,0]. Si la forme de masque n'est pas RotoBézier, le segment obtenu est un segment de droite.

Si la forme est un masque RotoBézier, toutes les valeurs sont ignorées et les tangentes sont calculées de façon automatique.

**Type**

Tableau de paires de valeurs à virgule flottante ; lecture/écriture.

**Attribut Shape vertices**

*shapeObject.value.vertices*

**Description**

Points d'ancrage de la forme. Définissez chaque point sous forme de tableau de deux valeurs à virgule flottante et rassemblez les paires de points dans un tableau comprenant l'ensemble des points. Par exemple :

```
myShape.vertices = [[0,0], [0,1], [1,1], [1,0]];
```

**Type**

Tableau de paires de valeurs à virgule flottante ; lecture/écriture.

## Objet ShapeLayer

`app.project.item(index).layer(index)`

### Description

L'objet ShapeLayer représente un calque de forme dans une composition. Vous pouvez le créer à l'aide de la méthode `addShape()` de l'objet `LayerCollection` ; voir la section « Méthode `LayerCollection` `addShape()` », page 94. Pour accéder à cet objet dans la collection de calques d'un élément, vous pouvez utiliser son numéro d'index ou encore une chaîne de nom.

- ShapeLayer est une sous-classe de `AVLayer`, qui est lui-même une sous-classe de `Layer`. Toutes les méthodes et tous les attributs de `AVLayer` et de `Layer` sont disponibles lorsque vous utilisez ShapeLayer. Voir les sections « Objet `Layer` », page 83 et « Objet `AVLayer` », page 39.

## Objet SolidSource

*app.project.item(index).mainSource.*

*app.project.item(index).proxySource.*

### Description

L'objet SolidSource représente une source de métrage de couleur unie.

- SolidSource est une sous-classe de FootageSource. Toutes les méthodes et tous les attributs de FootageSource, outre ceux énumérés ci-dessous, sont disponibles lorsque vous utilisez SolidSource. Voir la section « Objet FootageSource », page 67.

### Attributs

Attribut	Référence	Description
color	« Attribut SolidSource color », page 170	Couleur du solide.

### Attribut SolidSource color

*solidSource.color*

### Description

Couleur du solide, exprimée en valeurs rouge, vert et bleu.

### Type

Tableau de trois valeurs à virgule flottante, [R, V, B], comprises dans la plage [0.0..1.0] ; lecture/écriture.

## Objet System

system

### Description

L'objet System fournit l'accès aux attributs trouvés sur le système de l'utilisateur, tels que le nom d'utilisateur et le nom et la version du système d'exploitation. Disponible par l'intermédiaire de la variable globale `system`.

### Exemple

```
alert("Your OS is " + system.osName + " running version " + system.osVersion);
confirm("You are: " + system.userName + " running on " + system.machineName + ".");
```

### Attributs

Attribut	Référence	Description
userName	« Attribut System userName », page 172	Nom de l'utilisateur actuel.
machineName	« Attribut System machineName », page 172	Nom de l'ordinateur hôte.
osName	« Attribut System osName », page 172	Nom du système d'exploitation.
osVersion	« Attribut System osVersion », page 172	Version du système d'exploitation.

### Méthodes

Méthode	Référence	Description
callSystem()	« Méthode System callSystem() », page 171	Exécute une commande dans la ligne de commande du système d'exploitation.

### Méthode System callSystem()

```
system.callSystem (cmdLineToExecute);
```

### Description

Exécute une commande système comme si vous l'aviez tapée dans la ligne de commande du système d'exploitation. Renvoie, le cas échéant, toute réponse que le système génère suite à l'exécution de cette commande.

Sous Windows, vous pouvez lancer des commandes à l'aide de l'option `/c` pour la commande `cmd.exe`, en plaçant la commande à exécuter entre guillemets d'échappement (guillemets précédés d'une barre oblique inversée : `\"...\"`). Par exemple, le code suivant extrait l'heure actuelle et l'affiche à l'utilisateur :

```
var timeStr = system.callSystem("cmd.exe /c \"time /t\"");
alert("Current time is " + timeStr);
```

### Paramètres

cmdLineToExecute	Chaîne contenant la commande et ses paramètres.
------------------	---

### Renvoi

Résultat généré par la commande.

**Attribut System machineName**

system.machineName

**Description**

Nom de l'ordinateur sur lequel After Effects est installé.

**Type**

Chaîne ; lecture seule.

**Attribut System osName**

system.osName

**Description**

Nom du système d'exploitation sous lequel After Effects fonctionne.

**Type**

Chaîne ; lecture seule.

**Attribut System osVersion**

system.osVersion

**Description**

Version du système d'exploitation local actuel.

**Type**

Chaîne ; lecture seule.

**Attribut System userName**

system.userName

**Description**

Nom de l'utilisateur actuellement connecté sur le système.

**Type**

Chaîne ; lecture seule.

## Objet TextDocument

```
new TextDocument(docText)  
app.project.item(index).layer(index).property("Source Text").value
```

### Description

L'objet TextDocument stocke une valeur pour la propriété Source Text d'un TextLayer. Créez-le à l'aide du constructeur, en stipulant la chaîne à y introduire.

### Exemples

Ce script définit une valeur de texte source et affiche une alerte indiquant la nouvelle valeur :

```
var myTextDocument = new TextDocument("Happy Cake");  
myTextLayer.property("Source Text").setValue(myTextDocument);  
alert(myTextLayer.property("Source Text").value);
```

Ce script définit des valeurs d'image clé pour du texte qui affichent différents mots dans le temps :

```
var textProp = myTextLayer.property("Source Text");  
textProp.setValueAtTime(0, new TextDocument("Happy"));  
textProp.setValueAtTime(.33, new TextDocument("cake"));  
textProp.setValueAtTime(.66, new TextDocument("is"));  
textProp.setValueAtTime(1, new TextDocument("yummy!"));
```

### Attributs

Attribut	Référence	Description
text	« Attribut TextDocument text », page 173	Valeur Source Text du calque de texte.

### Attribut TextDocument text

*textDocument*.text

### Description

Valeur de texte pour la propriété Source Text du calque de texte.

### Type

Chaîne ; lecture/écriture.

## Objet TextLayer

`app.project.item(index).layer(index)`

### Description

L'objet TextLayer représente un calque de texte dans une composition. Vous pouvez le créer à l'aide de la méthode `addText` de l'objet `LayerCollection` ; voir la section « Méthode `LayerCollection addText()` », page 95. Pour accéder à cet objet dans la collection de calques d'un élément, vous pouvez utiliser son numéro d'index ou encore une chaîne de nom.

- TextLayer est une sous-classe de AVLayer, qui est lui-même une sous-classe de Layer. Toutes les méthodes et tous les attributs de AVLayer et de Layer sont disponibles lorsque vous utilisez TextLayer. Voir les sections « Objet Layer », page 83 et « Objet AVLayer », page 39.

### Propriétés AE

TextLayer ne définit pas d'attributs supplémentaires, mais possède, outre ceux hérités de AVLayer, les propriétés AE et les groupes de propriétés suivants :

Text

Source Text

Path Options

Path

Reverse Path

Perpendicular To Path

Force Alignment

First Margin

Last Margin

More Options

Anchor Point Grouping

Grouping Alignment

Fill & Stroke

Inter-Character Blending

Animators

### Attributs et propriétés non utilisés

Les propriétés `Time Remap` et `Motion Trackers`, héritées de AVLayer, ne sont pas applicables aux calques de texte. De même, les attributs relatifs à AVLayer ne peuvent pas être utilisés :

`canSetTimeRemapEnabled`

`timeRemapEnabled`

`trackMatteType`

`isTrackMatte`

`hasTrackMatte`

# Exemples

Cette section présente des exemples de scripts inclus sur le DVD accompagnant le produit ; elle explique leur rôle et décrit leur fonctionnement.

Loin d'être exhaustive, cette série d'exemples démontre cependant certaines des fonctionnalités de création de script les plus complexes. Elle présente également certaines constructions type de programmation JavaScript applicables à la création de script.

Pour obtenir davantage d'exemples fournis par Adobe ou d'autres utilisateurs d'After Effects, visitez le site Adobe Studio Exchange à l'adresse <http://share.studio.adobe.com>, puis choisissez Scripts dans la zone Adobe After Effects.

## Save and increment (enregistrement et incrémentation)

Le script `save_and_increment.jsx` enregistre automatiquement une nouvelle copie du projet After Effects ouvert et incrémente un numéro à trois chiffres inclus dans son nom pour le distinguer de ses versions précédentes.

*Remarque : bien que la majeure partie de la fonctionnalité de ce script ait été remplacée par la fonction d'enregistrement incrémentielle introduite dans After Effects 6.5, ce script reste inclus ici, car il utilise de façon efficace les conditionnels, les fonctions et l'objet `ExtendScript File`.*

Ce script effectue les tâches suivantes :

- Détermine si le projet ouvert a déjà été enregistré. Si le projet n'a pas été enregistré, affiche un message d'alerte demandant à l'utilisateur d'enregistrer le projet et s'arrête.
- Si le projet a déjà été enregistré au moins une fois, définit des variables pour le nom du fichier et le numéro et l'extension de fichier à y ajouter.
- Vérifie s'il existe un caractère de soulignement à une distance de quatre caractères de la fin du nom de fichier actuel. Dans ce cas, considère que l'incrémenteur a été exécuté précédemment et incrémente la chaîne numérique en cours, puis extrait le nom sans cette extension numérique.
- Une boucle d'incrémentation vérifie si le numéro est passé à deux ou trois chiffres (par exemple, si le numéro a atteint ou dépassé « `_010` » ou « `_100` ») et attribue un zéro pour chacun dans le cas contraire.
- Crée un nouveau fichier utilisant le nom et l'extension mis à jour et affiche un message d'alerte informant l'utilisateur du nouveau nom de fichier enregistré.
- Enregistre le projet sous le nouveau nom de fichier.

## Render named items (rendu des éléments nommés)

Le script `renderNamedItems.jsx` recherche dans le projet ouvert les compositions dont le nom comporte une chaîne de texte donnée et les envoie à la file d'attente de rendu.

Ce script effectue les tâches suivantes :

- Vérifie si une chaîne par défaut pour le rendu a déjà été définie dans les préférences utilisateur. Le cas échéant, cette dernière est définie en tant qu'invite utilisateur. Cette fonction est particulièrement pratique lorsque vous recherchez toujours la même chaîne (par exemple, « FINAL » ou « CURRENT »). Sinon, un nouveau `sectionName` et un nouveau `keyName` sont définis dans le fichier de préférences, ainsi qu'une valeur de pseudo-élément pour la chaîne qui sera entrée par l'utilisateur.
- Affiche un message pour demander à l'utilisateur quelle chaîne de texte utiliser.
- Recherche le texte entré par l'utilisateur dans le projet et vérifie que l'élément qui contient ce texte est une composition. Envoie toutes les compositions correspondantes à la file d'attente de rendu.
- Si l'utilisateur annule, le texte est indéfini. Sinon, enregistre le nouveau paramètre dans les préférences, en le convertissant en minuscules par souci de cohérence (bien que la recherche ne tienne pas compte de la casse).
- Affiche le panneau File d'attente de rendu au premier plan pour permettre à l'utilisateur de spécifier les emplacements d'enregistrement des nouveaux éléments de la file d'attente de rendu.

## New render locations (nouveaux emplacements de rendu)

Le script `newRenderLocations.jsx` permet à l'utilisateur de sélectionner des éléments en file d'attente de rendu et de leur attribuer une nouvelle destination de rendu.

Ce script effectue les tâches suivantes :

- Invite l'utilisateur à indiquer un nouveau dossier à utiliser comme destination de rendu.
- Vérifie que l'utilisateur a bien entré un nouvel emplacement (et n'a pas annulé), puis crée une boucle pour chaque élément de file d'attente de rendu sélectionné et pour chaque module de sortie qu'il contient.
- Si un élément est placé en file d'attente, attribue un nouveau nom et un nouvel emplacement au rendu en cours, et affiche un message d'alerte indiquant le nouveau chemin du fichier.

## Smart import (importation avancée)

Le script `smartImport.jsx` permet à l'utilisateur d'importer la totalité du contenu imbriqué d'un dossier simplement en le sélectionnant. Il tente de détecter si chaque élément est une image fixe, un métrage mobile ou une séquence d'images. L'utilisateur doit encore effectuer d'autres choix au moyen de boîtes de dialogue, comme le choix du calque d'une image multicalque (un fichier PSD, par exemple) à importer.

Ce script effectue les tâches suivantes :

- Invite l'utilisateur à indiquer un dossier dont le contenu doit être importé et vérifie que l'utilisateur choisit bien un dossier et n'annule pas le message.
- Définit une fonction, `processFolder()`, pour importer chaque fichier dans le dossier choisi à l'aide de plusieurs fonctions d'assistance.
- Définit une fonction d'assistance, `testForSequence()`, pour vérifier si un fichier donné fait partie d'une séquence. Celle-ci utilise des expressions régulières, lesquelles constituent un type spécial de JavaScript destiné à réduire le nombre d'étapes requises pour évaluer une chaîne.

La première vérifie la présence de chiffres séquentiels à tout endroit du nom de fichier. La suivante s'assure que les fichiers séquentiels ne sont pas d'un type ne pouvant pas être importé en tant que séquence (fichiers d'image mobiles). La fonction vérifie ensuite les fichiers adjacents pour savoir s'il existe une séquence. Cette vérification s'arrête à dix fichiers pour limiter le temps de traitement.

Si la recherche d'une chaîne numérique ne génère aucun résultat, le script considère qu'il n'y a pas de séquence d'images et vérifie la présence d'un tableau constitué de la chaîne correspondante et de son emplacement dans le nom de fichier.

Si tous les fichiers font partie d'une séquence numérique, le script conclut qu'il y a une séquence et renvoie le premier fichier de cette séquence.

- Définit une fonction d'assistance pour afficher des messages d'erreur en cas de problème avec tout fichier que le script tente d'importer.
- Définit une fonction d'assistance pour importer toute séquence d'images détectée à l'aide de `testForSequence()`. Le script contient une option placée en commentaire permettant de forcer l'ordre alphabétique dans les séquences. Pour forcer l'ordre alphabétique, annulez le statut de commentaire de la ligne `importOptions.forceAlphabetical = true`.
- Appelle la fonction principale, `processFolder()`.

## Render and e-mail (rendu et message électronique)

Le script `render_and_email.jsx` effectue le rendu de tous les éléments en file d'attente dans un projet ouvert et envoie un rapport par courrier électronique signalant le moment où le rendu s'est achevé. Il utilise deux autres scripts, à savoir : `email_methods.jsx` (pour envoyer le courrier électronique correctement) et `email_setup.jsx` (qui établit l'expéditeur, le destinataire et le serveur de messagerie).

Ce script effectue les tâches suivantes :

- Vérifie les conditions d'exécution du script. Un projet ouvert contenant au moins un élément en file d'attente est requis.
- Vérifie si les paramètres de courrier électronique sont déjà enregistrés dans les préférences. Sinon, le script `email_setup.jsx` est exécuté. L'utilisateur doit alors indiquer la passerelle de messagerie, ainsi que les adresses de l'expéditeur et du destinataire. (Si vous devez modifier des paramètres enregistrés, vous pouvez exécuter le script pour créer des paramètres venant remplacer les paramètres existants.)
- Effectue le rendu des éléments de la file d'attente de rendu.
- Une fois le rendu terminé, crée une chaîne de texte pour le message électronique. Cette chaîne contient l'heure de début du rendu, l'heure du rendu de chaque élément de la file d'attente et la durée totale du rendu.
- Envoie le message, en utilisant les paramètres (serveur, par exemple) du script `email_methods.jsx`.
- Affiche une erreur si, pour une raison quelconque, le message ne peut pas être envoyé.

Un script d'assistance, `email_methods.jsx`, crée un objet e-mail à l'aide de l'objet `ExtendScript Socket`. Pour plus d'informations sur cet utilitaire, consultez le manuel *Creative Suite 3 JavaScript Tools Guide*.

Un autre script d'assistance, `email_setup.jsx`, invite l'utilisateur à indiquer le nom du serveur, l'expéditeur et le destinataire du message électronique, lesquels sont enregistrés en tant que paramètres. Vous pouvez exécuter ce script de manière autonome chaque fois que vous souhaitez modifier les paramètres. Ce script illustre bien la façon de créer des paramètres enregistrés dans les préférences à des fins exclusives de création de script (et non pour la modification de paramètres de préférences After Effects existants).

## **Convert selected properties to markers (conversion de propriétés sélectionnées en repères)**

Le script `Convert Selected Properties to Markers.jsx` parcourt les propriétés de calque actuellement sélectionnées dans le panneau Montage et convertit la valeur de chaque propriété à chaque instant d'image en un signal de départ vidéo Flash de type événement inclus dans un repère.

Ce script ajoute pour chaque propriété sélectionnée un repère temporel de calque sur le calque au même instant que les images clés. Chaque repère est associé à un signal de départ vidéo Flash de type événement, lui-même associé à un paramètre dont le nom et la valeur correspondent à ceux de la propriété à cet instant. Si la propriété sélectionnée contient une expression, un repère est créé pour chaque image, avec échantillonnage des valeurs à chaque image.

*Remarque : ce script ne convertit pas les propriétés impliquant des types de valeurs complexes, telles que la propriété `Tracé d'un tracé de peinture`, la propriété `Courbes d'un effet Courbe` ou une propriété de dégradé.*

Lors du rendu de la composition en vidéo Flash, tous les repères contenant des données de signal de départ sont convertis en signaux de départ vidéo Flash.

# Liste des objets After Effects

Cette sortie de code récapitule tous les objets JavaScript (classes instanciables) et les types énumérés définis pour After Effects CS3.

```
=====
AlphaMode enum
-----
AlphaMode.IGNORE
AlphaMode.PREMULTIPLIED
AlphaMode.STRAIGHT
=====

Application object
-----

activate() no return
beginSuppressDialogs() no return
beginUndoGroup(string undoName) no return
buildName : string : readOnly
buildNumber : integer : readOnly
cancelTask(integer taskID) no return
endSuppressDialogs(boolean showAlert) no return
endUndoGroup() no return
endWatchFolder() no return
exitAfterLaunchAndEval : boolean : read/write
exitCode : integer : read/write
findMenuCommandId() returns integer
isProfessionalVersion : boolean : readOnly
isRenderEngine : boolean : readOnly
isUISuppressed : boolean : readOnly
isWatchFolder : boolean : readOnly
language : Language : readOnly
memoryInUse : number : readOnly
newProject() no return
open([File file]) returns Project
openTemplate(File fileToOpenWithTemplateSemantics) no return
parseSwatchFile(File swatchFile) returns SwatchObject
pauseWatchFolder(boolean doPause) no return
project : Project : readOnly
purge(PurgeTarget target) no return
quit() no return
saveProjectOnCrash : boolean : read/write
scheduleTask(stringToExecute, float delay, boolean repeat) returns taskID
setMemoryUsageLimits(float imageCachePercent, float maximumMemoryPercent) no return
setSavePreferencesOnQuit(boolean doSave) no return
settings : Settings : readOnly
```

```

version : string : readOnly
watchFolder(File file) no return
onError(string errorString, string severity) no return

```

```

=====
AVLayer object
-----

```

```

(integer propertyIndex) returns PropertyBase
(string propertyName) returns PropertyBase
active : boolean : readOnly
activeAtTime(float atTime) returns boolean
addProperty(string propertyName) returns PropertyBase
adjustmentLayer : boolean : read/write
applyPreset(string presetName) no return
audioActive : boolean : readOnly
audioActiveAtTime(float atTime) returns boolean
audioEnabled : boolean : read/write
autoOrient : AutoOrientType : read/write
blendingMode : BlendingMode : read/write
calculateTransformFromPoints(Array [top,left], Array [top,right], Array [bottom,right])
    returns Object with transform properties set
canAddProperty(string propertyName) returns boolean
canSetCollapseTransformation : boolean : readOnly
canSetEnabled : boolean : readOnly
canSetTimeRemapEnabled : boolean : readOnly
collapseTransformation : boolean : read/write
comment : string : read/write
containingComp : CompItem : readOnly
copyToComp(CompItem intoComp) no return
duplicate() returns AVLayer
effectsActive : boolean : read/write
elided : boolean : readOnly
enabled : boolean : read/write
frameBlending : boolean : readOnly
frameBlendingType : FrameBlendingType : read/write
guideLayer : boolean : read/write
hasAudio : boolean : readOnly
hasTrackMatte : boolean : readOnly
hasVideo : boolean : readOnly
height : float : readOnly
inPoint : float : read/write
index : integer : readOnly
isEffect : boolean : readOnly
isMask : boolean : readOnly
isModified : boolean : readOnly
isNameFromSource : boolean : readOnly
isNameSet : boolean : readOnly
isTrackMatte : boolean : readOnly
locked : boolean : read/write
matchName : string : readOnly
motionBlur : boolean : read/write
moveAfter(Layer otherLayer) no return

```

```

moveBefore(Layer otherLayer) no return
moveTo(integer index) no return
moveToBeginning() no return
moveToEnd() no return
name : string : read/write
nullLayer : boolean : readOnly
numProperties : integer : readOnly
outPoint : float : read/write
parent : Layer : read/write
parentProperty : PropertyGroup : readOnly
preserveTransparency : boolean : read/write
property(integer propertyIndex) returns PropertyBase
property(string propertyName) returns PropertyBase
propertyDepth : integer : readOnly
propertyGroup([integer countUp]) returns PropertyGroup
propertyType : PropertyType : readOnly
quality : LayerQuality : read/write
remove() no return
replaceSource(Item newSource, boolean fixExpressions) no return
selected : boolean : read/write
selectedProperties : Array of PropertyBase: readOnly
setParentWithJump(Layer newParent) no return
shy : boolean : read/write
solo : boolean : read/write
source : AVItem : readOnly
sourceRectAtTime(float atTime, boolean includeExtents)
    returns Object with float properties: top; left; width; height
startTime : float : read/write
stretch : float : read/write
threeDLayer : boolean : read/write
threeDPerChar : boolean : read/write
time : float : readOnly
timeRemapEnabled : boolean : read/write
trackMatteType : TrackMatteType : read/write
width : float : readOnly

```

```

=====
BlendingMode enum
-----

```

```

BlendingMode.ADD
BlendingMode.ALPHA_ADD
BlendingMode.CLASSIC_COLOR_BURN
BlendingMode.CLASSIC_COLOR_DODGE
BlendingMode.CLASSIC_DIFFERENCE
BlendingMode.COLOR
BlendingMode.COLOR_BURN
BlendingMode.COLOR_DODGE
BlendingMode.DANCING DISSOLVE
BlendingMode.DARKEN
BlendingMode.DARKER_COLOR
BlendingMode.DIFFERENCE
BlendingMode.DISSOLVE

```

BlendingMode.EXCLUSION  
BlendingMode.HARD\_LIGHT  
BlendingMode.HARD\_MIX  
BlendingMode.HUE  
BlendingMode.LIGHTEN  
BlendingMode.LIGHTER\_COLOR  
BlendingMode.LINEAR\_BURN  
BlendingMode.LINEAR\_DODGE  
BlendingMode.LINEAR\_LIGHT  
BlendingMode.LUMINESCENT\_PREMUL  
BlendingMode.LUMINOSITY  
BlendingMode.MULTIPLY  
BlendingMode.NORMAL  
BlendingMode.OVERLAY  
BlendingMode.PIN\_LIGHT  
BlendingMode.SATURATION  
BlendingMode.SCREEN  
BlendingMode.SILHOUETTE\_ALPHA  
BlendingMode.SILHOUETTE\_LUMA  
BlendingMode.SOFT\_LIGHT  
BlendingMode.STENCIL\_ALPHA  
BlendingMode.STENCIL\_LUMA  
BlendingMode.VIVID\_LIGHT

=====  
CloseOptions enum

-----  
CloseOptions.DO\_NOT\_SAVE\_CHANGES  
CloseOptions.PROMPT\_TO\_SAVE\_CHANGES  
CloseOptions.SAVE\_CHANGES  
=====

CompItem object

-----  
activeCamera : Layer : readOnly  
applyPreset(string presetName) no return  
bgColor : Array of float : read/write  
comment : string : read/write  
displayStartTime : float : read/write  
draft3d : boolean : read/write  
duplicate() returns CompItem  
duration : float : read/write  
footageMissing : boolean : readOnly  
frameBlending : boolean : read/write  
frameDuration : float : read/write  
frameRate : float : read/write  
hasAudio : boolean : readOnly  
hasVideo : boolean : readOnly  
height : integer : read/write  
hideShyLayers : boolean : read/write  
id : integer : readOnly  
layer(integer layerIndex) returns Layer  
layer(string layerName) returns Layer

```

layer(Layer otherLayer, integer relativeIndex) returns Layer
layers : LayerCollection: readOnly
motionBlur : boolean : read/write
name : string : read/write
numLayers : integer : readOnly
parentFolder : FolderItem : read/write
pixelAspect : float : read/write
preserveNestedFrameRate : boolean : read/write
preserveNestedResolution : boolean : read/write
proxySource : FootageSource : readOnly
remove() no return
renderer : string : read/write
renderers : Array of string: readOnly
resolutionFactor : Array of integer : read/write
selected : boolean : read/write
selectedLayers : Array of Layer : readOnly
selectedProperties : Array of PropertyBase: readOnly
setProxy(File proxyFile) no return
setProxyToNone() no return
setProxyWithPlaceholder(string name, integer width, integer height, float frameRate, float duration)
    no return
setProxyWithSequence(File proxyFile, boolean forceAlphabetical) no return
setProxyWithSolid(ArrayOfFloat color, string name, integer width, integer height,
    float pixelAspecRatio) no return
shutterAngle : integer : read/write
shutterPhase : integer : read/write
time : float : read/write
typeName : string : readOnly
useProxy : boolean : read/write
usedIn : Array of CompItem : readOnly
width : integer : read/write
workAreaDuration : float : read/write
workAreaStart : float : read/write
=====
FieldSeparationType enum
-----
FieldSeparationType.LOWER_FIELD_FIRST
FieldSeparationType.OFF
FieldSeparationType.UPPER_FIELD_FIRST
=====
FileSource object
-----
alphaMode : AlphaMode : read/write
conformFrameRate : float : read/write
displayFrameRate : float : readOnly
fieldSeparationType : FieldSeparationType : readOnly
file : File : readOnly
guessAlphaMode() no return
guessPullDown(PullDownMethod pullDownMethod) no return
hasAlpha : boolean : readOnly
highQualityFieldSeparation : boolean : read/write

```

invertAlpha : boolean : read/write  
isStill : boolean : readOnly  
loop : integer : read/write  
missingFootagePath : string : readOnly  
nativeFrameRate : float : readOnly  
premulColor : Array of float : read/write  
reload() no return  
removePulldown : PulldownPhase : readOnly

=====  
FolderItem object  
-----

comment : string : read/write  
id : integer : readOnly  
item(integer itemIndex) returns Item  
items : ItemCollection : readOnly  
name : string : read/write  
numItems : integer : readOnly  
parentFolder : FolderItem : read/write  
remove() no return  
selected : boolean : read/write  
typeName : string : readOnly

=====  
FootageItem object  
-----

comment : string : read/write  
duration : float : readOnly  
file : File : readOnly  
footageMissing : boolean : readOnly  
frameDuration : float : readOnly  
frameRate : float : readOnly  
hasAudio : boolean : readOnly  
hasVideo : boolean : readOnly  
height : integer : read/write  
id : integer : readOnly  
mainSource : FootageSource : readOnly  
name : string : read/write  
parentFolder : FolderItem : read/write  
pixelAspect : float : read/write  
proxySource : FootageSource : readOnly  
remove() no return  
replace(File proxyFile) no return  
replaceWithPlaceholder(string name, integer width, integer height, float frameRate, float duration)  
no return  
replaceWithSequence(File proxyFile, boolean forceAlphabetical) no return  
replaceWithSolid(ArrayOfFloat color, string name, integer width, integer height, float pixelAspecRatio)  
no return  
selected : boolean : read/write  
setProxy(File proxyFile) no return  
setProxyToNone() no return  
setProxyWithPlaceholder(string name, integer width, integer height, float frameRate, float duration)  
no return

```
setProxyWithSequence(File proxyFile, boolean forceAlphabetical) no return
setProxyWithSolid(ArrayOfFloat color, string name, integer width, integer height,
    float pixelAspectRatio) no return
time : float : readOnly
typeName : string : readOnly
useProxy : boolean : read/write
usedIn : Array of CompItem : readOnly
width : integer : read/write
```

=====  
ImportAsType enum

```
-----
ImportAsType.COMP
ImportAsType.COMP_CROPPED_LAYERS
ImportAsType.FOOTAGE
ImportAsType.PROJECT
=====
```

ImportOptions object

```
-----
new ImportOptions(File fileToImport) returns ImportOptions
canImportAs(ImportAsType asType) returns boolean
file : File : read/write
forceAlphabetical : boolean : read/write
importAs : ImportAsType : read/write
sequence : boolean : read/write
=====
```

ItemCollection object

```
-----
addComp(string name, integer width, integer height, float pixelAspectRatio, float duration,
    float frameRate) returns CompItem
addFolder(string name) returns FolderItem
=====
```

KeyframeEase object

```
-----
new KeyframeEase(float speed, float influence) returns KeyframeEase
influence : float : read/write
speed : float : read/write
=====
```

KeyframeInterpolationType enum

```
-----
KeyframeInterpolationType.BEZIER
KeyframeInterpolationType.HOLD
KeyframeInterpolationType.LINEAR
=====
```

Language enum

```
-----
Language.ENGLISH
Language.FRENCH
Language.GERMAN
Language.ITALIAN
Language.JAPANESE
Language.SPANISH
```

=====  
CameraLayer object  
-----

(integer propertyIndex) returns PropertyBase  
(string propertyName) returns PropertyBase  
active : boolean : readOnly  
activeAtTime(float atTime) returns boolean  
addProperty(string propertyName) returns PropertyBase  
adjustmentLayer : boolean : readOnly  
applyPreset(string presetName) no return  
autoOrient : AutoOrientType : read/write  
canAddProperty(string propertyName) returns boolean  
canSetEnabled : boolean : readOnly  
comment : string : read/write  
containingComp : CompItem : readOnly  
copyToComp(CompItem intoComp) no return  
duplicate() returns CameraLayer  
elided : boolean : readOnly  
enabled : boolean : read/write  
hasVideo : boolean : readOnly  
inPoint : float : read/write  
index : integer : readOnly  
isEffect : boolean : readOnly  
isMask : boolean : readOnly  
isModified : boolean : readOnly  
isNameSet : boolean : readOnly  
locked : boolean : read/write  
matchName : string : readOnly  
moveAfter(Layer otherLayer) no return  
moveBefore(Layer otherLayer) no return  
moveTo(integer index) no return  
moveToBeginning() no return  
moveToEnd() no return  
name : string : read/write  
nullLayer : boolean : readOnly  
numProperties : integer : readOnly  
outPoint : float : read/write  
parent : Layer : read/write  
parentProperty : PropertyGroup : readOnly  
property(integer propertyIndex) returns PropertyBase  
property(string propertyName) returns PropertyBase  
propertyDepth : integer : readOnly  
propertyGroup([integer countUp]) returns PropertyGroup  
propertyType : PropertyType : readOnly  
remove() no return  
selected : boolean : read/write  
selectedProperties : Array of PropertyBase: readOnly  
setParentWithJump(Layer newParent) no return  
shy : boolean : read/write  
solo : boolean : read/write  
startTime : float : read/write

```

stretch : float : read/write
time : float : readOnly
=====
LayerCollection object
-----
add(AVItem theItem,
 [float duration]) returns AVLayer
addCamera(string name,
 ArrayOfFloat centerPoint) returns Layer
addLight(string name,
 ArrayOfFloat centerPoint) returns Layer
addNull([float duration]) returns AVLayer
addShape() returns Layer
addSolid(ArrayOfFloat color, string name, integer width, integer height, float pixelAspectRatio,
 [float duration]) returns AVLayer
addText([TextDocument textDoc]) returns AVLayer
addText(string text) returns AVLayer
byName(string name) returns Layer
precompose(ArrayOfInteger layerIndices, string name, [boolean moveAllAttributes])
 returns CompItem
=====
LayerQuality enum
-----
LayerQuality.BEST
LayerQuality.DRAFT
LayerQuality.WIREFRAME
=====
LogType enum
-----
LogType.ERRORS_AND_PER_FRAME_INFO
LogType.ERRORS_AND_SETTINGS
LogType.ERRORS_ONLY
=====
MarkerValue object
-----
new MarkerValue(string comment, [string chapter], [string url], [string frameTarget])
 returns MarkerValue
chapter : string : read/write
comment : string : read/write
cuePointName : string : read/write
eventCuePoint : boolean : read/write
frameTarget : string : read/write
getParameters() returns object with properties set
setParameters(Object keyValuePairs) no return
url : string : read/write
=====
MaskMode enum
-----
MaskMode.ADD
MaskMode.DARKEN
MaskMode.DIFFERENCE

```

MaskMode.INTERSECT  
 MaskMode.LIGHTEN  
 MaskMode.NONE  
 MaskMode.SUBTRACT

=====  
 MaskMotionBlur enum  
 -----

MaskMotionBlur.OFF  
 MaskMotionBlur.ON  
 MaskMotionBlur.SAME\_AS\_LAYER

=====  
 MaskPropertyGroup object  
 -----

(integer propertyIndex) returns PropertyBase  
 (string propertyName) returns PropertyBase  
 active : boolean : readOnly  
 addProperty(string propertyName) returns PropertyBase  
 canAddProperty(string propertyName) returns boolean  
 canSetEnabled : boolean : readOnly  
 color : Array of float : read/write  
 duplicate() returns MaskPropertyGroup  
 elided : boolean : readOnly  
 enabled : boolean : readOnly  
 inverted : boolean : read/write  
 isEffect : boolean : readOnly  
 isMask : boolean : readOnly  
 isModified : boolean : readOnly  
 locked : boolean : read/write  
 maskMode : MaskMode : read/write  
 maskMotionBlur : MaskMotionBlur : read/write  
 matchName : string : readOnly  
 moveTo(integer index) no return  
 name : string : read/write  
 numProperties : integer : readOnly  
 parentProperty : PropertyGroup : readOnly  
 property(integer propertyIndex) returns PropertyBase  
 property(string propertyName) returns PropertyBase  
 propertyDepth : integer : readOnly  
 propertyGroup([integer countUp]) returns PropertyGroup  
 propertyIndex : integer : readOnly  
 propertyType : PropertyType : readOnly  
 remove() no return  
 rotoBezier : boolean : read/write  
 selected : boolean : read/write

=====  
 OMCollection object  
 -----

add() returns OutputModule

=====  
 OutputModule object  
 -----

```

applyTemplate(string templateName) no return
file : File : read/write
name : string : readOnly
postRenderAction : PostRenderAction : read/write
remove() no return
saveAsTemplate(string templateName) no return
templates : Array of string: readOnly

```

=====

PlaceholderSource object

```

-----
alphaMode : AlphaMode : read/write
conformFrameRate : float : read/write
displayFrameRate : float : readOnly
fieldSeparationType : FieldSeparationType : read/write
guessAlphaMode() no return
guessPullDown(PullDownMethod pullDownMethod) no return
hasAlpha : boolean : readOnly
highQualityFieldSeparation : boolean : read/write
invertAlpha : boolean : read/write
isStill : boolean : readOnly
loop : integer : read/write
nativeFrameRate : float : readOnly
premulColor : Array of float : read/write
removePullDown : PullDownPhase : read/write

```

=====

PostRenderAction enum

```

-----
PostRenderAction.IMPORT
PostRenderAction.IMPORT_AND_REPLACE_USAGE
PostRenderAction.NONE
PostRenderAction.SET_PROXY

```

=====

Project object

```

-----
activeItem : Item : readOnly
autoFixExpressions(oldText,
newText) no return
bitsPerChannel : integer : read/write
close(CloseOptions closeOptions) returns boolean
consolidateFootage() returns integer
displayStartFrame : integer : read/write
file : File : readOnly
importFile(ImportOptions importOptions) returns Item
importFileWithDialog() returns ArrayOfItem
importPlaceholder(string itemName, integer itemWidth, integer itemHeight, float frameRate,
float duration) returns FootageItem
item(integer itemIndex) returns Item
items : ItemCollection : readOnly
linearBlending : boolean : read/write
numItems : integer : readOnly
reduceProject(ArrayOfItem itemsToPreserve) returns integer

```

```

removeUnusedFootage() returns integer
renderQueue : RenderQueue : readOnly
rootFolder : FolderItem : readOnly
save(File toFile) returns boolean
saveWithDialog() returns boolean
selection : Array of Item : readOnly
showWindow(boolean doShow) no return
timecodeBaseType : TimecodeBaseType : read/write
timecodeDisplayType : TimecodeDisplayType : read/write
timecodeFilmType : TimecodeFilmType : read/write
timecodeNTSCDropFrame : boolean : read/write
transparencyGridThumbnails : boolean : read/write

```

```

=====
Property object
-----

```

```

active : boolean : readOnly
addKey(float atTime) returns integer
canSetEnabled : boolean : readOnly
canSetExpression : boolean : readOnly
canVaryOverTime : boolean : readOnly
duplicate() returns Property
elided : boolean : readOnly
enabled : boolean : readOnly
expression : string : read/write
expressionEnabled : boolean : read/write
expressionError : string : readOnly
hasMax : boolean : readOnly
hasMin : boolean : readOnly
isEffect : boolean : readOnly
isInterpolationTypeValid(KeyframeInterpolationType type) returns boolean
isMask : boolean : readOnly
isModified : boolean : readOnly
isSpatial : boolean : readOnly
isTimeVarying : boolean : readOnly
keyInInterpolationType(integer keyIndex) returns KeyframeInterpolationType
keyInSpatialTangent(integer keyIndex) returns ArrayOfFloat
keyInTemporalEase(integer keyIndex) returns ArrayOfKeyframeEase
keyOutInterpolationType(integer keyIndex) returns KeyframeInterpolationType
keyOutSpatialTangent(integer keyIndex) returns ArrayOfFloat
keyOutTemporalEase(integer keyIndex) returns ArrayOfKeyframeEase
keyRoving(integer keyIndex) returns boolean
keySelected(integer keyIndex) returns boolean
keySpatialAutoBezier(integer keyIndex) returns boolean
keySpatialContinuous(integer keyIndex) returns boolean
keyTemporalAutoBezier(integer keyIndex) returns boolean
keyTemporalContinuous(integer keyIndex) returns boolean
keyTime(integer keyIndex) returns float
keyTime(string markerName) returns float
keyValue(integer keyIndex) returns type-stored-in-property
keyValue(string markerName) returns type-stored-in-property
matchName : string : readOnly

```

```

moveTo(integer index) no return
name : string : readOnly
nearestKeyIndex(float atTime) returns integer
numKeys : integer : readOnly
parentProperty : PropertyGroup : readOnly
propertyDepth : integer : readOnly
propertyGroup([integer countUp]) returns PropertyGroup
propertyIndex : integer : readOnly
propertyType : PropertyType : readOnly
propertyValueType : PropertyValue : readOnly
remove() no return
removeKey(integer keyIndex) no return
selected : boolean : read/write
selectedKeys : Array of integer : readOnly
setInterpolationTypeAtKey(integer keyIndex, KeyframeInterpolationType inType,
    [KeyframeInterpolationType outType]) no return
setRovingAtKey(integer keyIndex, boolean isRoving) no return
setSelectedAtKey(integer keyIndex, boolean isSelected) no return
setSpatialAutoBezierAtKey(integer keyIndex, boolean isAutoBezier) no return
setSpatialContinuousAtKey(integer keyIndex, boolean isContinuous) no return
setSpatialTangentsAtKey(integer keyIndex, ArrayOfFloat inTangent, [ArrayOfFloat outTangent])
    no return
setTemporalAutoBezierAtKey(integer keyIndex, boolean isAutoBezier) no return
setTemporalContinuousAtKey(integer keyIndex, boolean isContinuous) no return
setTemporalEaseAtKey(integer keyIndex, ArrayOfKeyframeEase inEase,
    [ArrayOfKeyframeEase outEase]) no return
setValue(type-stored-in-property newValue) no return
setValueAtKey(integer keyIndex, type-stored-in-property newValue) no return
setValueAtTime(float atTime, type-stored-in-property newValue) no return
setValuesAtTimes(ArrayOfFloat atTimes, ArrayOf-type-stored-in-property newValues) no return
unitsText : string : readOnly
value : type-stored-in-property: readOnly
valueAtTime(float atTime, bool preExpression) returns type-stored-in-property
=====

```

#### PropertyGroup object

```

-----
(integer propertyIndex) returns PropertyBase
(string propertyName) returns PropertyBase
active : boolean : readOnly
addProperty(string propertyName) returns PropertyBase
canAddProperty(string propertyName) returns boolean
canSetEnabled : boolean : readOnly
duplicate() returns PropertyGroup
elided : boolean : readOnly
enabled : boolean : readOnly
isEffect : boolean : readOnly
isMask : boolean : readOnly
isModified : boolean : readOnly
matchName : string : readOnly
moveTo(integer index) no return
name : string : readOnly

```

```
numProperties : integer : readOnly
parentProperty : PropertyGroup : readOnly
property(integer propertyIndex) returns PropertyBase
property(string propertyName) returns PropertyBase
propertyDepth : integer : readOnly
propertyGroup([integer countUp]) returns PropertyGroup
propertyIndex : integer : readOnly
propertyType : PropertyType : readOnly
remove() no return
selected : boolean : read/write
```

=====  
PropertyType enum

-----  
PropertyType.INDEXED\_GROUP  
PropertyType.NAMED\_GROUP  
PropertyType.PROPERTY  
=====

PropertyValueType enum

-----  
PropertyValueType.COLOR  
PropertyValueType.CUSTOM\_VALUE  
PropertyValueType.LAYER\_INDEX  
PropertyValueType.MARKER  
PropertyValueType.MASK\_INDEX  
PropertyValueType.NO\_VALUE  
PropertyValueType.OneD  
PropertyValueType.SHAPE  
PropertyValueType.TEXT\_DOCUMENT  
PropertyValueType.ThreeD  
PropertyValueType.ThreeD\_SPATIAL  
PropertyValueType.TwoD  
PropertyValueType.TwoD\_SPATIAL  
=====

PullDownPhase enum

-----  
PullDownPhase.OFF  
PullDownPhase.SSWWW  
PullDownPhase.SWWWS  
PullDownPhase.SWWWW\_24P\_ADVANCE  
PullDownPhase.WSSWW  
PullDownPhase.WSWWW\_24P\_ADVANCE  
PullDownPhase.WWSSW  
PullDownPhase.WWSWW\_24P\_ADVANCE  
PullDownPhase.WWWSS  
PullDownPhase.WWWSW\_24P\_ADVANCE  
PullDownPhase.WWWWS\_24P\_ADVANCE  
=====

PullDownMethod enum

-----  
PullDownMethod.ADVANCE\_24P  
PullDownMethod.PULLDOWN\_3\_2

```
=====
PurgeTarget enum
-----
PurgeTarget.ALL_CACHES
PurgeTarget.IMAGE_CACHES
PurgeTarget.SNAPSHOT_CACHES
PurgeTarget.UNDO_CACHES
=====

RenderQueue object
-----

item(integer itemIndex) returns RenderQueueItem
items : RQItemCollection : readOnly
numItems : integer : readOnly
pauseRendering(boolean doPause) no return
render() no return
rendering : boolean : readOnly
showWindow(boolean doShow) no return
stopRendering() no return
=====

RenderQueueItem object
-----

applyTemplate(string templateName) no return
comp : CompItem : readOnly
duplicate() returns RenderQueueItem
elapsedSeconds : float : readOnly
logType : LogType : read/write
numOutputModules : integer : readOnly
outputModule(integer outputModuleIndex) returns OutputModule
outputModules : OMCollection : readOnly
remove() no return
render : boolean : read/write
saveAsTemplate(string templateName) no return
skipFrames : integer : read/write
startTime : float : readOnly
status : RQItemStatus : readOnly
templates : Array of string: readOnly
timeSpanDuration : float : read/write
timeSpanStart : float : read/write
onStatusChanged() no return
=====

RQItemCollection object
-----

add(CompItem compToAdd) returns RenderQueueItem
=====

RQItemStatus enum
-----

RQItemStatus.DONE
RQItemStatus.ERR_STOPPED
RQItemStatus.NEEDS_OUTPUT
RQItemStatus.QUEUED
RQItemStatus.RENDERING
```

```
RQItemStatus.UNQUEUED
RQItemStatus.USER_STOPPED
RQItemStatus.WILL_CONTINUE
```

=====

Settings object

```
-----
getSetting(string sectionName, string sectionKey) returns string
haveSetting(string sectionName, string sectionKey) returns boolean
saveSetting(string sectionName, string sectionKey, string newValue) no return
```

=====

Shape object

```
-----
new Shape() returns Shape
closed : boolean : read/write
inTangents : Array of float[2] : read/write
outTangents : Array of float[2] : read/write
vertices : Array of float[2] : read/write
```

=====

SolidSource object

```
-----
alphaMode : AlphaMode : read/write
color : Array of float : read/write
conformFrameRate : float : readOnly
displayFrameRate : float : readOnly
fieldSeparationType : FieldSeparationType : readOnly
guessAlphaMode() no return
guessPulldown(PulldownMethod pulldownMethod) no return
hasAlpha : boolean : readOnly
highQualityFieldSeparation : boolean : readOnly
invertAlpha : boolean : read/write
isStill : boolean : readOnly
loop : integer : readOnly
nativeFrameRate : float : readOnly
premulColor : Array of float : read/write
removePulldown : PulldownPhase : readOnly
```

=====

System object

```
-----
callSystem(string cmdLineToExecute) returns outputOfCommandAsString
machineName : string : readOnly
osName : string : readOnly
osVersion : string : readOnly
userName : string : readOnly
```

=====

TextDocument object

```
-----
new TextDocument(string text) returns TextDocument
text : string : read/write
```

=====

TimecodeBaseType enum

TimecodeBaseType.AUTO  
TimecodeBaseType.FPS100  
TimecodeBaseType.FPS24  
TimecodeBaseType.FPS25  
TimecodeBaseType.FPS30  
TimecodeBaseType.FPS48  
TimecodeBaseType.FPS50  
TimecodeBaseType.FPS60

=====  
TimecodeDisplayType enum

-----  
TimecodeDisplayType.FEET\_AND\_FRAMES  
TimecodeDisplayType.FRAMES  
TimecodeDisplayType.TIMECODE

=====  
TimecodeFilmType enum

-----  
TimecodeFilmType.MM16  
TimecodeFilmType.MM35

=====  
TrackMatteType enum

-----  
TrackMatteType.ALPHA  
TrackMatteType.ALPHA\_INVERTED  
TrackMatteType.LUMA  
TrackMatteType.LUMA\_INVERTED  
TrackMatteType.NO\_TRACK\_MATTE

=====  
AutoOrientType enum

-----  
AutoOrientType.ALONG\_PATH  
AutoOrientType.CAMERA\_OR\_POINT\_OF\_INTEREST  
AutoOrientType.NO\_AUTO\_ORIENT

=====  
FrameBlendingType enum

-----  
FrameBlendingType.FRAME\_MIX  
FrameBlendingType.NO\_FRAME\_BLEND  
FrameBlendingType.PIXEL\_MOTION