

Adobe® InCopy® 2022 スクリプティング — お読みください

本ドキュメントには、Adobe InCopy 2022 のスクリプティングに関する次の情報が記載されています。

- ◆ InCopy のスクリプティングマニュアルの概要 ([1 ページの「InCopy のスクリプティングマニュアル」](#)を参照)
- ◆ スクリプトの実行方法 ([2 ページの「スクリプトの実行」](#)を参照)
- ◆ InCopy のサンプルスクリプトのリストと各スクリプトの簡単な説明 ([2 ページの「サンプルスクリプト」](#)を参照)
- ◆ InCopy のスクリプティングに関する既知の問題 ([4 ページの「InCopy のスクリプティングに関する既知の問題」](#)を参照)

InCopy のスクリプティングについて詳しくは、InDesign デベロッパーマニュアルページ (<https://www.adobe.io/apis/creativecloud/indesign.html>) (英語) を参照してください。または、InCopy Scripting ユーザーフォーラム (http://forums.adobe.com/community/incopy/incopy_scripting) (英語) にアクセスしてください。

InCopy のスクリプティングに関する最新情報については、InDesign デベロッパーマニュアルページに掲載されているこのファイルの最新版を参照してください。

InCopy のスクリプティングマニュアル

InCopy のスクリプティングマニュアルは、上述の InDesign デベロッパーマニュアルページからダウンロードできます。InCopy のスクリプティングマニュアルセットには、次のマニュアルが含まれています。

- ◆ 『Adobe InCopy スクリプティングガイド』 (AppleScript、JavaScript、VBScript の各バージョン) — InCopy のスクリプティングに関する基本的なチュートリアルのほか、より高度なスクリプティング技術に関する情報が含まれています。チュートリアル全スクリプトは 1 つの ZIP アーカイブに含まれているため、PDF からスクリプトをコピーして貼り付ける必要はありません (テキストに記載されているスクリプトのほとんどは、特定のプロパティ、メソッド、または技法を説明するための断片的なスクリプトです)。
- ◆ 『JavaScript Tools and Features』 — ExtendScript Toolkit を使用した JavaScript 開発、ScriptUI を使用したユーザーインターフェイスの作成、ファイルオブジェクトやフォルダーオブジェクトの使用など、ExtendScript 言語 (アドビ版 JavaScript) に特有の機能について説明しています。

このリリースには『Scripting Reference』の PDF はありません。スクリプト編集アプリケーションに含まれているオブジェクトモデルビューアを代わりに使用してください (『Adobe InCopy スクリプティングガイド』を参照)。

InCopy のサンプルスクリプトはデフォルトでインストールされ、スクリプトパネル (ウィンドウ/ユーティリティ/スクリプト) に表示されます。

スクリプティングマニュアルのスクリプトのインストール

サンプルスクリプトだけでなく、スクリプティングマニュアルに記載されているすべてのスクリプト（断片的なスクリプト）は InDesign デベロッパーマニュアルページからダウンロード可能です。

スクリプト ZIP アーカイブをダウンロードしてアーカイブからスクリプトを展開したら、使用する言語（AppleScript、JavaScript または VBScript）のフォルダーを、Scripts Panel フォルダーに移動します。スクリプトのインストールについて詳しくは、『Adobe InCopy スクリプティングガイド』を参照してください。

スクリプトの実行

スクリプトを実行するには、スクリプトパネルでスクリプト名をダブルクリックします。

Windows の場合は、VBScript（.vbs というファイル拡張子）または JavaScript（.jsx というファイル拡張子）を実行できます。Mac OS の場合は、AppleScript（.applescript というファイル拡張子）または JavaScript を実行できます。

サンプルスクリプト

サンプルスクリプトを初めて実行する前に作業内容を保存することを忘れないでください。

重要な InCopy ドキュメント上で実際にスクリプトを使用する前に、サンプルスクリプトの動作について理解できるように、サンプルスクリプトを試してみてください。

InCopy には、次のサンプルスクリプトが含まれています。

スクリプト名	説明	詳細参照ページ
CreateCharacterStyle	選択したテキストに基づいて文字スタイルを定義します。	3
ExportAllStories	ドキュメント内のすべてのストーリーをテキストファイルに書き出します。	3
FindChangeByList	タブ区切りテキストファイルを読み取り、一般的なテキストの検索と置換操作を行います。	3
SortParagraphs	選択範囲内の段落をアルファベット順に並べ替えます。	3
TabUtilities	選択したテキストにタブストップおよびインデントを適用します。	4

スクリプトの説明

この節では、各サンプルスクリプトについて簡単に説明します。

CreateCharacterStyle

選択したテキストに基づいて文字スタイルを定義します。

説明：

- ◆ 選択されたオブジェクトに対する処理
- ◆ 文字スタイルの作成
- ◆ テキストオブジェクトからのテキスト書式属性の取得
- ◆ 文字スタイルプロパティの適用

ExportAllStories

ドキュメント内のすべてのストーリーをテキストファイルに書き出します。

説明：

- ◆ テキストの書き出し
- ◆ JavaScript のファイル／フォルダーオブジェクトおよびメソッド（JavaScript のみ）
- ◆ ID 属性に基づいたファイル名の作成
- ◆ ユーザーインターフェイスの作成

FindChangeByList

タブ区切りテキストファイルを読み取り、一般的なテキストの検索と置換操作を行います。

詳しくは、スクリプトの先頭または FindChangeList.txt ファイルの先頭を参照してください。
FindChangeList.txt ファイルに独自の検索と置換操作を追加したり、独自のファイルを作成したりすることができます。

説明：

- ◆ ファイルとフォルダーの操作
- ◆ 実行中のスクリプトからの相対的なファイル／フォルダーの検索
- ◆ テキストの検索と置換方法（text、grep、glyph）
- ◆ テキストファイルからのタブ区切りテキストデータの読み込み

SortParagraphs

選択範囲内の段落をアルファベット順に並べ替えます。

説明：

- ◆ テキストオブジェクトの move メソッド
- ◆ 簡単なバブルソート

TabUtilities

選択したテキストにタブストップおよびインデントを適用します。

説明：

- ◆ タブストップとインデントの操作
- ◆ テキストオブジェクトからのページ位置の取得
- ◆ カーソルを含むテキスト列の取得
- ◆ ユーザーインターフェイスの作成

InCopy のスクリプティングに関する既知の問題

JavaScript スタートアップスクリプトの場所

ユーザースタートアップスクリプトは、ExtendScript エンジン初期化スクリプトの場所ではなく（ここに格納されたスクリプトはエンジンの初期化時に実行されます）、InCopy スタートアップスクリプトの場所（ここに格納されたスクリプトはアプリケーションの起動のたびに一度実行されます）に保存する必要があります。

InCopy の起動時にスクリプトを自動実行させるには、InCopy フォルダー内の Scripts フォルダーの下にあるスタートアップ用のフォルダーである startup scripts フォルダーにスクリプトファイルを保存します（このフォルダーがない場合は作成してください）。

InCopy の外部で実行したスクリプトでパーシスタント ExtendScript エンジンを作成できない（JavaScript のみ）

『Adobe InCopy スクリプティングガイド：JavaScript』の第2章「スクリプティング」で説明しているとおり、ExtendScript スクリプトでは ExtendScript エンジンのパーシスタントインスタンスを作成することができます。パーシスタントエンジンで定義された関数や変数は、そのエンジンで実行されるその他のスクリプトから使用することができます。パーシスタント ExtendScript エンジンを作成するには、InCopy のスクリプトパネルからスクリプトを実行する必要があります。ExtendScript Toolkit からスクリプトを実行したり、別のアプリケーションから BridgeTalk 経由でスクリプトを実行したりした場合、パーシスタントエンジンは作成されません。

イベント伝達中に追加または削除されたイベントリスナーが W3C の仕様どおりに処理されない

『W3C Document Object Model (DOM) Level 2 Events Specification』

(<http://www.w3.org/TR/DOM-Level-2-Events/Overview.html> を参照) に、次のような記述があります。

「If an EventListener is added to an EventTarget while it is processing an event, it will not be triggered by the current actions but may be triggered during a later stage of event flows such as the bubbling phase. (参考訳：イベント処理中にイベントターゲットにイベントリスナーが追加された場合、そのイベントリスナーは現在のアクションでは始動しませんが、イベントフローの後の段階（例えば、バブリング段階）で始動する可能性があります。）」

さらに、次のような記述もあります。

「If an EventListener is removed from an EventTarget while it is processing an event, it will not be triggered by the current actions. EventListeners can never be invoked after being removed. EventListeners can never be invoked after being removed. (参考訳：イベント処理中にイベントターゲットからイベントリスナーが削除された場合、そのイベントリスナーは現在のアクションで始動しません。イベントリスナーは削除すると一切呼び出されません。)」

InCopy のスクリプティングでは、イベント伝達中にイベントターゲットに追加されたイベントリスナーは、そのイベント中は始動しません。イベント伝達中にイベントターゲットから削除されたイベントリスナーは、そのイベントで始動します（つまり、イベントリスナーはイベント処理が終了してから削除されます）。

ExtendScript Toolkit に InCopy のスクリプトのリストが表示されない (Mac OS のみ)

ExtendScript Toolkit では、デフォルトで起動時にターゲットアプリケーションが指定されていません。つまり、ExtendScript Toolkit のスクリプトパネル (InCopy のではない) の使用可能スクリプトのリストに、InCopy のスクリプトが表示されないということです。ターゲットアプリケーションを InCopy に設定すると、ExtendScript Toolkit のスクリプトパネルに InCopy のスクリプトが表示されます。