

# Scripting Methods

## setAEMCredentials

Sets the credentials for the target AEM server.

Target object:

Application

Arguments:

userID:

String

Required

AEM user id to use.

password:

String

Required

AEM password to use.

Example:

```
app.setAEMCredentials("user", "password");
```

## setHTTPDomains

Description:

Set the HTTP place white listed domain(s) and/or sub-domain(s).

Target object:

Application

Arguments:

domains:

String

Required

White listed domains/sub-domains placed HTTP links are allowed to target. Multiple domains/sub-domains may be specified, separated by spaces.

Example:

```
app.setHTTPDomains("author-condenastqa.adobecqms.com wikimedia.org");
```

## clearHTTPDomains

### Description:

Clears the HTTP place white list. Clearing the white list results in all domains being allowed.

### Target object:

Application

### Arguments:

N/A:

### Example:

```
app.clearHTTPDomains();
```

## placeHTTPLink

### Description:

Place an HTTP link.

### Target object:

Document

### Arguments:

#### URLs:

String

Required

HTTP URLs to the assets to be placed. Multiple HTTP URLs may be specified, separated by spaces.

#### showOptionsDialog:

Boolean

Optional default=False

Denotes whether to display the import options dialog.

#### createLink:

Boolean

Optional default=False

Denotes whether to create a link for the placed text assets.

#### replaceSelected:

Boolean

Optional default=True

Denotes whether to replace the selected item. The argument is ignored if multiple URLs are specified, or multiple page items are selected.

### Example:

```
var doc = app.activeDocument;  
doc.placeHTTPLink("https://author-condenastqa.adobecqms.com/asset/357403c8-fc78-49ea-  
b735-efa60ba6b594//boot1.jpg");
```

**updateHTTPLink**

## Description:

Updates the specified HTTP link.

## Target object:

Link

## Arguments:

N/A:

## Returns:

Link

## Note:

The only difference between this method and InDesign's update script method is that this method will always update the link regardless of whether it is not out-of-date or not.

## Example:

```
var doc = app.activeDocument;  
var docLinks = doc.links;  
  
docLinks[0].updateHTTPLink();
```

**relinkHTTPLink**

## Description:

Relink an HTTP link.

## Target object:

Link

## Arguments:

URL:

URI

Required

HTTP URL of the asset to relink to.

## Note:

This method is similar to InDesign's relink script method, but it fixes some URI encoding issues that the relink method has.

## Example:

```
var doc = app.activeDocument;  
var docLinks = doc.links;  
var link = docLinks[0];
```

```
link.relinkHTTPLink("https://author-condenastqa.adobecqms.com/asset/357403c8-fc78-49ea-  
b735-efa60ba6b594//boot1.jpg");
```

**clearHTTPCache**

## Description:

Clears/empties the Connector HTTP cache.

## Target object:

Application

## Arguments:

N/A:

## Example:

```
app.clearHTTPCache();
```

**trimHTTPCache**

## Description:

Trims the Connector HTTP cache to no more than 75% of the cache size limit. If no cache size limit is set, the call is a no op.

## Target object:

Application

## Arguments:

N/A:

## Example:

```
app.trimHTTPCache();
```

# Scripting Properties

## uriType

The link's URI type.

Target object:

Link

Returns:

LinkUriType Enum

file

VersionCue

Buzzword

HTTP

HTTPS

other

Example:

```
var doc = app.activeDocument;  
var docLinks = doc.links;  
var link = docLinks[0];  
  
var linkType = link.uriType;
```

## uriPath

The link's URI or a file path, depending upon the link's URI type.

Target object:

Link

Returns:

File for file LinkUriType

String for all other LinkUriType

Example:

```
var link = app.activeDocument.links[0];  
var linkType = link.uriType;  
  
if (linkType == LinkUriType.file) {  
    var filePath = link.uriPath;  
    alert(filePath);  
}  
else if (linkType == LinkUriType.HTTP) {  
    var uriPath = link.uriPath;  
    alert(uriPath);  
}
```

**autoUpdateLink**

Denotes whether the link will auto update when the linked asset changes.

Target object:

Link

Returns:

Boolean

Example:

```
var link = app.activeDocument.links[0];
var autoUpdate = link.autoUpdateLink;

if (!autoUpdate) {
    link.autoUpdateLink = true;
}
```

**cacheSize**

Current size in bytes of the Connector HTTP cache.

Target object:

Application

Returns:

Integer

Example:

```
var size = app.cacheSize;
alert("Cache size: " + size);
```

**cacheDir**

The directory location of the Connector HTTP cache. When set, the subfolders SiliconConnector/Caches are always appended to the given path.

Target object:

Application

Returns:

File

Example:

```
var dir = app.cacheDir;
alert("Cache location: " + dir);

app.cacheDir = "/Users/userName/Documents";
```

**cacheLimit**

The maximum size limit of the Connector HTTP cache in megabytes. Set to 0 for no limit.

Make sure to set the limit to a reasonable size in relation to the size of the assets being linked via HTTP. Making the limit too small could result in frequent downloads of a linked assets data. If necessary, the cache will be trimmed when the size limit is changed.

Target object:

Application

Returns:

Integer (min: 0, max: 100000 MB)

Example:

```
var limit = app.cacheLimit;

if (limit == 0) {
  alert("Cache size is not limited!");
}
else {
  alert("Cache size limit: " + limit + " MB");
}

app.cacheLimit = 500;
```

**linkStatusInterval**

The minimum number of seconds between HTTP link status updates.

A shorter interval results in users being notified of changes to the link asset more quickly. However, a shorter interval will also result in more frequent server request for status information.

Target object:

Application

Returns:

Integer (min: 3, max: 300)

Example:

```
var interval = app.linkStatusInterval;
alert("Link status update interval: " + interval + " seconds");

app.linkStatusInterval = 20;
```