
**Document management — ECMAScript
for PDF —**

Part 1:
Use of ISO 32000-2 (PDF 2.0)

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
<https://standards.iteh.ai/catalog/standards/sist/5684a307-2e89-4bf1-a943-ab0cfa0cc85c/iso-prf-21757-1>

PROOF / ÉPREUVE



iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
<https://standards.iteh.ai/catalog/standards/sist/5684a307-2e80-4bf1-a943-ab0cfa0cc85c/iso-prf-21757-1>



COPYRIGHT PROTECTED DOCUMENT

© ISO 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

	Page
Foreword.....	ix
Introduction.....	x
1 Scope.....	1
2 Normative references.....	1
3 Terms and definitions.....	1
4 Notation.....	1
5 Syntax.....	2
5.1 General.....	2
5.2 Method arguments.....	2
6 Paths.....	2
7 Safe path.....	2
8 Privileged context.....	3
9 Privileged versus non-privileged context.....	3
10 ECMAScript API.....	3
10.1 General.....	3
10.2 Annotation.....	3
10.2.1 General.....	3
10.2.2 Annotation types.....	4
10.2.3 Annotation properties.....	5
10.2.4 Annotation methods.....	13
10.2.5 Annotation examples.....	15
10.3 AnnotRichMedia.....	17
10.3.1 General.....	17
10.3.2 AnnotRichMedia properties.....	17
10.4 Annot3D.....	17
10.4.1 General.....	17
10.4.2 Annot3D properties.....	17
10.5 app.....	18
10.5.1 General.....	18
10.5.2 app properties.....	18
10.5.3 app methods.....	20
10.6 Bookmark.....	30
10.6.1 General.....	30
10.6.2 Bookmark properties.....	30
10.6.3 Bookmark methods.....	31
10.6.4 Bookmark Examples.....	32
10.7 Certificate.....	33
10.7.1 General.....	33
10.7.2 Certificate properties.....	34
10.8 color.....	36
10.8.1 General.....	36
10.8.2 color arrays.....	36
10.8.3 color properties.....	36
10.8.4 color methods.....	37
10.9 collection.....	38
10.9.1 General.....	38
10.9.2 collection properties.....	38
10.9.3 collection methods.....	38
10.10 collectionField.....	40
10.10.1 General.....	40

10.10.2	collectionField properties.....	40
10.11	Data.....	41
10.11.1	General.....	41
10.11.2	Data properties.....	42
10.11.3	Data methods.....	42
10.12	Dialog.....	43
10.12.1	General.....	43
10.12.2	Dialog methods.....	43
10.13	Doc.....	44
10.13.1	General.....	44
10.13.2	Doc properties.....	45
10.13.3	Doc methods.....	49
10.14	Embedded PDF.....	89
10.14.1	General.....	89
10.14.2	Embedded PDF properties.....	90
10.14.3	Embedded PDF methods.....	91
10.15	Error.....	91
10.15.1	General.....	91
10.15.2	Error properties.....	92
10.15.3	Error methods.....	92
10.16	event.....	93
10.16.1	General.....	93
10.16.2	Event type/name combinations.....	93
10.16.3	Document Event Processing.....	99
10.16.4	Form event processing.....	99
10.16.5	event properties.....	100
10.17	Field.....	105
10.17.1	General.....	105
10.17.2	Field versus widget attributes.....	106
10.17.3	Field properties.....	106
10.17.4	Field methods.....	118
10.18	FullScreen.....	135
10.18.1	General.....	135
10.18.2	FullScreen properties.....	135
10.19	global.....	136
10.19.1	General.....	136
10.19.2	Creating global properties.....	136
10.19.3	Deleting global properties.....	137
10.19.4	Global object security policy.....	137
10.19.5	global object methods.....	137
10.20	HostContainer.....	138
10.20.1	General.....	138
10.20.2	HostContainer properties.....	139
10.20.3	HostContainer methods.....	140
10.21	Icon.....	141
10.21.1	General.....	141
10.21.2	icon Properties.....	141
10.22	Link.....	141
10.22.1	General.....	141
10.22.2	Link properties.....	141
10.22.3	Link methods.....	142
10.23	Net.....	142
10.23.1	General.....	142
10.23.2	Net properties.....	142
10.23.3	Net methods.....	144
10.24	OCG.....	146
10.24.1	General.....	146
10.24.2	OCG properties.....	146

10.24.3	OCG methods	147
10.25	PrintParams	148
10.25.1	General	148
10.25.2	PrintParams properties	148
10.26	RDN	151
10.26.1	General	151
10.26.2	RDN properties	152
10.27	ReadStream	152
10.27.1	General	152
10.27.2	ReadStream methods	152
10.28	security	153
10.28.1	General	153
10.28.2	security constants	153
10.28.3	security Properties	153
10.28.4	security Methods	154
10.29	SecurityHandler	157
10.29.1	General	157
10.29.2	SecurityHandler properties	157
10.29.3	SecurityHandler methods	160
10.30	SecurityPolicy	163
10.30.1	General	163
10.30.2	SecurityPolicy properties	163
10.31	SignatureInfo	163
10.31.1	General	163
10.31.2	SignatureInfo Base properties	163
10.31.3	SignatureInfo object public key security handler properties	165
10.31.4	Modification Detection and Prevention (MDP) Values	168
10.32	SOAP	168
10.32.1	General	168
10.32.2	SOAP properties	169
10.32.3	SOAP methods	169
10.33	Span	181
10.33.1	General	181
10.33.2	Span properties	181
10.34	Template	183
10.34.1	General	183
10.34.2	Template properties	183
10.34.3	Template methods	183
10.35	Thermometer	184
10.35.1	General	184
10.35.2	Thermometer properties	184
10.35.3	Thermometer methods	185
10.36	this	185
10.36.1	General	185
10.36.2	Variable and function name conflicts	186
10.37	util	186
10.37.1	General	186
10.37.2	util methods	186
11	ECMAScript 3D API	193
11.1	General	193
11.1.1	Basic Objects	193
11.1.2	Scene object	193
11.1.3	Canvas object	193
11.1.4	Runtime object	194
11.1.5	Resource objects	194
11.2	Event handlers	194
11.2.1	General	194
11.2.2	CameraEvent	194

11.2.3	KeyEvent	194
11.2.4	MouseEvent	194
11.2.5	RenderEvent	195
11.2.6	ScrollWheelEvent	195
11.2.7	SelectionEvent	195
11.2.8	TimeEvent	195
11.2.9	ToolEvent	195
12	Object overview	196
12.1	General	196
12.2	Animation	196
12.2.1	General	196
12.2.2	Animation properties	196
12.3	Background	196
12.3.1	General	196
12.3.2	Background object properties	196
12.3.3	Background object methods	196
12.4	BoundingBox	197
12.4.1	General	197
12.4.2	BoundingBox properties	197
12.5	Camera	197
12.5.1	General	197
12.5.2	Camera properties	198
12.5.3	Camera methods	199
12.6	CameraEvent	199
12.6.1	General	199
12.6.2	CameraEvent properties	199
12.7	CameraEventHandler	200
12.7.1	General	200
12.7.2	CameraEventHandler methods	200
12.8	Canvas	201
12.8.1	General	201
12.8.2	Canvas properties	201
12.8.3	Canvas methods	201
12.9	ClippingPlane	202
12.9.1	General	202
12.9.2	ClippingPlane Methods	202
12.10	Color	202
12.10.1	General	202
12.10.2	Color properties	202
12.10.3	Color methods	202
12.11	HitInfo	203
12.11.1	General	203
12.11.2	HitInfo properties	204
12.12	Host	204
12.12.1	General	204
12.13	Image	204
12.13.1	General	204
12.13.2	Image properties	204
12.13.3	Image methods	204
12.14	KeyEvent	205
12.14.1	General	205
12.14.2	KeyEvent properties	205
12.15	KeyEventHandler	207
12.15.1	General	207
12.15.2	KeyEventHandler methods	208
12.16	Light	208
12.16.1	General	208
12.16.2	Light properties	208

12.17	Material	209
12.17.1	General	209
12.17.2	Material properties	209
12.18	Matrix4x4	210
12.18.1	General	210
12.18.2	Matrix4x4 Properties	210
12.18.3	Matrix4x4 Methods	211
12.19	Mesh	219
12.19.1	General	219
12.19.2	Mesh properties	219
12.19.3	Mesh methods	219
12.20	MouseEvent	220
12.20.1	General	220
12.20.2	MouseEvent properties	220
12.21	MouseEventHandler	221
12.21.1	General	221
12.21.2	MouseEventHandler properties	221
12.21.3	MouseEventHandler methods	222
12.22	Node	222
12.22.1	General	222
12.22.2	Node properties	223
12.22.3	Node methods	223
12.23	Quaternion	224
12.23.1	General	224
12.23.2	Quaternion methods	224
12.24	RenderEvent	226
12.24.1	General	226
12.24.2	RenderEvent properties	226
12.25	RenderEventHandler	226
12.25.1	General	226
12.25.2	RenderEventHandler methods	226
12.26	Resource	228
12.26.1	General	228
12.26.2	Resource properties	228
12.26.3	Resource methods	228
12.27	Runtime	228
12.27.1	General	228
12.27.2	Runtime properties	228
12.27.3	Runtime methods	230
12.28	Scene	235
12.28.1	General	235
12.28.2	Scene methods	238
12.29	SceneObject	240
12.29.1	General	240
12.30	SceneObjectList	240
12.30.1	General	240
12.30.2	SceneObjectList methods	240
12.31	ScrollWheelEvent	241
12.31.1	General	241
12.31.2	ScrollWheelEvent	241
12.32	ScrollWheelEventHandler	242
12.32.1	General	242
12.32.2	ScrollWheelEventHandler methods	242
12.33	SelectionEvent	242
12.33.1	General	242
12.33.2	SelectionEvent properties	242
12.34	SelectionEventHandler	243
12.34.1	General	243

12.34.2	SelectionEventHandler methods	243
12.35	StateEvent	243
12.35.1	General	243
12.35.2	StateEvent properties	243
12.36	StateEventHandler	244
12.36.1	General	244
12.36.2	StateEventHandler methods	244
12.37	Texture	244
12.37.1	General	244
12.37.2	Texture properties	244
12.38	TimeEvent	245
12.38.1	General	245
12.38.2	TimeEvent properties	245
12.39	TimeEventHandler	245
12.39.1	General	245
12.39.2	TimeEventHandler methods	245
12.40	ToolEvent	246
12.40.1	General	246
12.40.2	ToolEventHandler properties	246
12.41	ToolEventHandler	246
12.41.1	General	246
12.41.2	ToolEventHandler methods	246
12.42	Vector3	247
12.42.1	General	247
12.42.2	Vector3 methods	247
12.43	View	252
12.43.1	General	252
12.43.2	View properties	252
Bibliography		253

iTeh STANDARD PREVIEW
 (standards.iteh.ai)
 Full standard:
<https://standards.iteh.ai/catalog/standards/sist/5684a307-2e80-4bf1-a943-ab0cfa0cc85c/iso-ptr-21757-1>

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 171, *Document management applications*, Subcommittee SC 2, *Document file formats, EDMS systems and authenticity of information*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document specifies a set of ECMAScript object types which define the properties and methods that can be used in ECMAScript scripts embedded in PDF documents to automate and interact with the containing PDF document and the PDF objects within such files.

The goal is to enable the implementation of ECMAScript processors within a broad range of PDF Processors to provide interoperable scripting and automation of PDF documents. This functionality includes the following features, among others:

- processing forms within the document;
- batch processing collections of PDF documents;
- developing and maintaining online collaboration schemes;
- communicating with local databases.

Certain properties and methods that may be discoverable through ECMAScript's introspection facilities are not documented here. Undocumented properties and methods should not be used.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
<https://standards.iteh.ai/catalog/standards/sist/5684a307-2e80-4bf1-a943-ab0cfa0cc85c/iso-prf-21757-1>

Document management — ECMAScript for PDF —

Part 1: Use of ISO 32000-2 (PDF 2.0)

1 Scope

This document defines a set of ECMAScript objectives for automating and interacting with PDF documents and the contents of such documents.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 32000-2, *Document management — Portable Document Format — Part 2: PDF 2.0*

ISO/IEC 22275:2018, *Information technology — Programming languages, their environments, and system software interfaces — ECMAScript® Specification Suite*

ISO/IEC 22537:2006, *Information technology — ECMAScript for XML (E4X) specification*

3 Terms and definitions

For the purposes of this document, the terms and definitions in ISO 32000-2 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

3.1

ECMAScript

means of reference to ISO 22275 and ISO 22537

3.2

SHA

Secure Hash Algorithms, means of reference to FIPS 180-4

4 Notation

ECMAScript objects and properties and other predefined names are written in bold font; values, as well as key terms of interest are written in italic font. Some names can also be used as values, depending on the context, and so the styling of the content will be context specific. Methods, functions, and variables are written in fixed-width font.

EXAMPLE 1 The allowed values for the **projectionType** property are *perspective* and *orthographic*.

Token characters used to delimit objects and describe the structure of PDF files, as defined in ISO 32000-2:—, 7.2.1, may be identified by their ISO/IEC 646 character name written in upper case in bold font followed by a parenthetic two digit hexadecimal character value with the suffix “h”.

EXAMPLE 2 **CARRIAGE RETURN** (0Dh).

Text string characters, as defined by ISO 32000-2:—, 7.9.2, may be identified by their ISO/IEC 10646-1 character name written in uppercase in bold font followed by a parenthetic four digit hexadecimal character code value with the prefix “U+”.

EXAMPLE 3 **EN SPACE** (U+2002).

A property labeled as **R** (read-only) is one whose value cannot be set. An object labeled as **R** (read-only) is one whose reference cannot be modified, though the object itself can be set and its properties may be modified. Unless otherwise indicated, all properties and objects labeled with **R/W** have read/write access.

5 Syntax

5.1 General

Some ECMAScript objects are static objects that can be used as is and must be spelled as indicated. For example, the `app` object represents the ECMAScript application. There is only one such object and it must be spelled `app` (case-sensitive).

Other objects are dynamic objects that can be assigned to a variable. For example, a `Doc` object may be obtained and assigned to a variable:

```
var myDoc = app.newDoc();
```

In this example, `myDoc` can access all methods and properties of the `Doc` object. For example:

```
myDoc.closeDoc();
```

5.2 Method arguments

Many of the ECMAScript methods accept either a list of arguments, as is customary in ECMAScript, or a single object argument with properties that contain the arguments. For example, these two calls are equivalent:

```
app.alert( "Multimedia", 3);
```

and

```
app.alert({ cMsg: "Multimedia", nIcon: 3});
```

NOTE The ECMAScript methods defined in support of multimedia do not accept these two argument formats interchangeably. Use the exact argument format described for each method.

6 Paths

Several methods take *device-independent paths* as arguments. See ISO 32000-2 for details about the device-independent path format.

7 Safe path

Developers of PDF Processor software implementing ECMAScript support are encouraged to support the concept of a *safe path* for ECMAScript methods that write data to the local hard drive based on a path passed to it by one of its parameters.

A path cannot point to a system critical folder, for example a root, windows or system directory. A path is also subject to other unspecified tests.

For many methods, the file name must have an extension appropriate to the type of data that is to be saved. Some methods may have a no-overwrite restriction. These additional restrictions are noted in the documentation.

Generally, when a path is judged to not be safe, a `NotAllowedError` exception is thrown (see [10.15 Error](#)) and the method fails.

8 Privileged context

An application context in which the application has the permissions necessary to do something on behalf of the current user that is normally restricted. Such permission (or privilege) could be granted by executing a method in a specific way (through the console or batch process), by some PDF property, or because the document was signed with a digital signature trusted by the user. For example, trusting a document certifier's certificate for executing ECMAScript creates a privileged context which enables the ECMAScript to run where it otherwise would not.

9 Privileged versus non-privileged context

Some ECMAScript methods, identified with a `[Security]` note, have security restrictions. These methods can be executed only in a privileged context, which includes console, batch and application initialization events. All other events (for example, page open and mouse-up events) are considered non-privileged.

The description of each security-restricted method indicates the events during which the method can be executed.

10 ECMAScript API

10.1 General

This section is a complete reference to the PDF extensions to ECMAScript, its objects, methods, and properties. The section is organized alphabetically by object name.

More information regarding the ECMAScript core can be found in ISO/IEC 22537.

10.2 Annotation

10.2.1 General

This object represents a PDF *markup annotation* (See ISO 32000-2:—, “12.5.6.2 Markup annotations” for more details). Annotations can be created through ECMAScript by using the `Doc` object method `addAnnot`.

Before an annotation can be accessed, it must be bound to an ECMAScript variable through a `Doc` object method such as `getAnnot`:

```
var a = this.getAnnot(0, "Important");
```

The script can then manipulate the annotation named “Important” on page 1 (0-based page numbering system) by means of the variable `a`. For example, the following code first stores the type of the annotation represented by `a` in the variable `thetype`, then changes the author to “John Q. Public”.

```
var thetype = a.type;           // read property
a.author = "John Q. Public";    // write property
```