

## Runtime keys

Technical  
Report



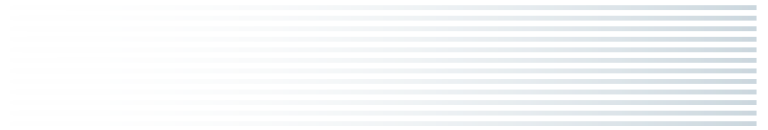
is the registered trademark of Ecma International



**COPYRIGHT PROTECTED DOCUMENT**

<b>Contents</b>		<b>Page</b>
<b>1</b>	<b>Scope</b> .....	<b>1</b>
<b>2</b>	<b>Terms and definitions</b> .....	<b>1</b>
<b>3</b>	<b>Runtime key structure</b> .....	<b>1</b>
<b>3.1</b>	<b>Key format</b> .....	<b>1</b>
<b>3.2</b>	<b>Key uniqueness</b> .....	<b>2</b>
<b>3.3</b>	<b>Key naming guidelines</b> .....	<b>2</b>
<b>3.4</b>	<b>Runtime existence validation</b> .....	<b>2</b>
<b>3.5</b>	<b>Runtime metadata</b> .....	<b>2</b>
<b>4</b>	<b>Canonical source for runtime keys</b> .....	<b>2</b>
<b>5</b>	<b>Governance and modification process</b> .....	<b>5</b>
<b>5.1</b>	<b>Proposing new runtime key entries</b> .....	<b>5</b>
<b>5.1.1</b>	<b>Proposal process</b> .....	<b>5</b>
<b>5.1.2</b>	<b>Expert review</b> .....	<b>5</b>
<b>5.2</b>	<b>Modifying entries</b> .....	<b>5</b>
<b>5.2.1</b>	<b>Metadata modification</b> .....	<b>5</b>
<b>5.2.2</b>	<b>Key immutability</b> .....	<b>5</b>
<b>5.2.3</b>	<b>Deprecation</b> .....	<b>5</b>
<b>5.3</b>	<b>Publication cadence</b> .....	<b>6</b>
<b>Annex A (normative)</b>	<b>Machine-readable canonical source</b> .....	<b>7</b>
<b>A.1</b>	<b>JSON schema</b> .....	<b>7</b>
<b>A.2</b>	<b>Accessing the source data</b> .....	<b>7</b>
<b>Annex B (informative)</b>	<b>Example usage</b> .....	<b>9</b>
<b>B.1</b>	<b>package.json example</b> .....	<b>9</b>
<b>B.2</b>	<b>Configuration File Example</b> .....	<b>10</b>
	<b>Bibliography</b> .....	<b>11</b>
	<b>Software License</b> .....	<b>13</b>





## Introduction

This Technical Report defines a canonical source for identifiers of unique [ECMAScript runtime](#) environments, known as *runtime keys*.

This Ecma Technical Report was developed by Technical Committee 55 and was adopted by the General Assembly of June 2026

**ALTERNATIVE COPYRIGHT NOTICE AND COPYRIGHT LICENSE**

© 2026 Ecma International

By obtaining and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission under Ecma's copyright to copy, modify, prepare derivative works of, and distribute this work, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the work or portions thereof, including modifications:

(i) The full text of this COPYRIGHT NOTICE AND COPYRIGHT LICENSE in a location viewable to users of the redistributed or derivative work.

(ii) Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the Ecma alternative copyright notice should be included.

(iii) Notice of any changes or modifications, through a copyright statement on the document such as "This document includes material copied from or derived from [Runtime keys](https://github.com/WinterTC55/runtime-keys) <https://github.com/WinterTC55/runtime-keys>.

Copyright © Ecma International."

**Disclaimers**

THIS WORK IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE DOCUMENT WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the work without specific, written prior permission. Title to copyright in this work will at all times remain with copyright holders.

# Runtime keys

## 1 Scope

**Runtime keys** provide a consistent and predictable mechanism for identifying **ECMAScript runtimes** in a variety of contexts, including but not limited to project configuration files, package manifests, conditional exports, and runtime detection mechanisms. This document focuses on providing informative identifiers primarily for web servers, though other ECMAScript server environments may find it useful. These keys are not intended to be used in web browsers, which should be referenced by other mechanisms such as the browserslist project.

This Report defines:

- the canonical source of **runtime keys** and associated metadata,
- guidance for future **ECMAScript runtimes** to submit **runtime keys**, and
- TC55's governance process for adding and modifying **runtime keys** to the Report.

**NOTE 1** This Report does not define *how* **runtime keys** should be used by tools or libraries. The purpose of this Report is to prevent conflicts and provide a reliable, authoritative source of runtime identifiers.

**NOTE 2** Inclusion in this Report does not imply that the specified runtime is fully conformant with any Ecma specification, including the WinterTC Minimum Common API. Inclusion does not imply endorsement of any kind.

## 2 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

### 2.1

#### **ECMAScript runtime**

implementation of the ECMA-262 ECMAScript language specification

### 2.2

#### **runtime key**

unique string identifier that represents a specific **ECMAScript runtime** environment

## 3 Runtime key structure

### 3.1 Key format

A **runtime key** satisfies all the following:

- It is a string value.
- It is usable in common configuration formats such as JSON and YAML.
- It meaningfully represents the associated runtime environment.
- It is simple and unambiguous.

**NOTE** Future versions of this Report may define a formal grammar for allowable characters in **runtime keys**.

### 3.2 Key uniqueness

A [runtime key](#) included in this Report will not conflict with the following:

- any other existing [runtime key](#) in this Report, or
- any existing Browserslist [browser](#) <<https://github.com/browserslist/browserslist?tab=readme-ov-file#browsers>> entries.

NOTE This explicitly excludes references to *server* runtimes within Browserslist.

### 3.3 Key naming guidelines

[Runtime keys](#) are subject to expert review by TC55. Acceptable keys will:

- avoid being too similar to existing keys,
- avoid similarity to common English words or offensive terms,
- avoid being too generic or similar to general terminology such as "Web Runtimes", "Edge Runtimes", or "JavaScript Runtimes", and
- demonstrate reasonable specificity and clarity.

### 3.4 Runtime existence validation

The runtime environment represented by a key must actually exist, whether as open source, source-available, or proprietary software. Entries in this Report exist to identify real runtimes, not to reserve names for future projects.

NOTE A runtime should demonstrate "proof of use" before being added to the Report. Examples include usage in `package.json` exports fields, published documentation, or active community adoption.

### 3.5 Runtime metadata

Each [runtime key](#) entry includes the following metadata:

- **Organization** (required): The organization or individual responsible for the runtime
- **Name** (required): The human-readable name of the runtime
- **Key** (required): The unique string identifier
- **Description** (required): A brief description of the runtime
- **Website** (optional): The primary website for the runtime
- **Repository** (optional): The source code repository URL, if available
- **Deprecated** (required): A boolean indicating whether the key is deprecated

## 4 Canonical source for runtime keys

The following table lists [runtime keys](#) submitted to TC55 and reviewed by the Committee, sorted alphabetically by organization.

NOTE This section is generated from the machine-readable source data (see [A](#)).

**Table 1 — Registered Runtime Keys**

Key	Organization / Name	Description, Website, Repository
<b>andromeda</b>	Andromeda / Andromeda	A modern, and secure JavaScript & TypeScript runtime built in Rust, powered by the Nova engine and Oxc. Provides zero-config TypeScript, rich Web APIs (Canvas, Web Crypto, Storage, HTTP, SQLite), and aims for WinterTC compliance. <a href="https://tryandromeda.dev">Website</a> <https://tryandromeda.dev> <a href="https://github.com/tryandromeda/andromeda">Repository</a> <https://github.com/tryandromeda/andromeda>
<b>arvancloud</b>	Arvancloud / Edge Computing	The JavaScript runtime that powers Arvancloud Edge Computing. <a href="https://www.arvancloud.ir/en/products/edge-computing">Website</a> <https://www.arvancloud.ir/en/products/edge-computing>
<b>azion</b>	Azion / Edge Functions	Azion Edge Functions for ultra-low latency, edge-native applications, built with open standards for secure, high-performance serverless computing. <a href="https://www.azion.com/en/products/edge-functions/">Website</a> <https://www.azion.com/en/products/edge-functions/>
<b>bun</b>	Oven / Bun	Incredibly fast JavaScript runtime, bundler, transpiler and package manager - all in one. <a href="https://bun.sh/">Website</a> <https://bun.sh/> <a href="https://github.com/oven-sh/bun">Repository</a> <https://github.com/oven-sh/bun>
<b>convex</b>	Convex / Convex Runtime	The JavaScript runtime used to execute deterministic Query and Mutation Functions as well as nondeterministic Action Functions in the Convex backend platform. <a href="https://docs.convex.dev/functions/runtimes#default-convex-runtime">Website</a> <https://docs.convex.dev/functions/runtimes#default-convex-runtime> <a href="https://github.com/get-convex/convex-backend">Repository</a> <https://github.com/get-convex/convex-backend>
<b>deno</b>	Deno Land / Deno	A modern runtime for JavaScript and TypeScript. <a href="https://deno.com">Website</a> <https://deno.com> <a href="https://github.com/denoland/deno">Repository</a> <https://github.com/denoland/deno>
<b>edge-light</b>	Vercel / Edge Light	Developing, testing, and defining the runtime Web APIs for Edge infrastructure. <a href="https://edge-runtime.vercel.app/">Website</a> <https://edge-runtime.vercel.app/> <a href="https://github.com/vercel/edge-runtime">Repository</a> <https://github.com/vercel/edge-runtime>
<b>edge-routine</b>	Alibaba Cloud / edge-routine	The JavaScript/Webassembly runtime that powers Alibaba Cloud edge-routine. <a href="https://www.alibabacloud.com/help/en/dynamic-route-for-cdn/latest/er-overview">Website</a> <https://www.alibabacloud.com/help/en/dynamic-route-for-cdn/latest/er-overview>
<b>electron</b>	OpenJS Foundation / Electron	Build cross-platform desktop apps with JavaScript, HTML, and CSS. <a href="https://www.electronjs.org/">Website</a> <https://www.electronjs.org/> <a href="https://github.com/electron/electron">Repository</a> <https://github.com/electron/electron>
<b>fastly</b>	Fastly / JavaScript on Compute@Edge	JavaScript runtime for Fastly Compute@Edge. <a href="https://developer.fastly.com/learning/compute/javascript/">Website</a> <https://developer.fastly.com/learning/compute/javascript/> <a href="https://github.com/fastly/js-compute-runtime">Repository</a> <https://github.com/fastly/js-compute-runtime>
<b>kiesel</b>	Kiesel / Kiesel	A JavaScript engine and runtime written in Zig. <a href="https://kiesel.dev/">Website</a> <https://kiesel.dev/> <a href="https://codeberg.org/kiesel-js/runtime">Repository</a> <https://codeberg.org/kiesel-js/runtime>
<b>lagon</b>	Lagon / Lagon Runtime	Lagon is an open-source runtime and platform that allows developers to run TypeScript and JavaScript Functions at the Edge. <a href="https://lagon.app">Website</a> <https://lagon.app> <a href="https://github.com/lagonapp/lagon">Repository</a> <https://github.com/lagonapp/lagon>

**Table 1 — Registered Runtime Keys** *(continued)*

Key	Organization / Name	Description, Website, Repository
<b>moddable</b>	Moddable / Moddable SDK	Open source runtime for resource-constrained embedded devices using standard JavaScript and TypeScript. Supports standard ECMA-419 APIs. <a href="https://www.moddable.com/">Website</a> <https://www.moddable.com/> <a href="https://github.com/Moddable-OpenSource/moddable">Repository</a> <https://github.com/Moddable-OpenSource/moddable>
<b>netlify</b>	Netlify / Edge Functions	Edge Functions connect the Netlify platform and workflow with an open runtime standard at the network edge. <a href="https://docs.netlify.com/edge-functions/overview/">Website</a> <https://docs.netlify.com/edge-functions/overview/> <a href="https://github.com/netlify/edge-functions">Repository</a> <https://github.com/netlify/edge-functions>
<b>node</b>	OpenJS Foundation / Node.js	Node.js is an open-source, cross-platform JavaScript runtime environment. <a href="https://nodejs.org">Website</a> <https://nodejs.org> <a href="https://github.com/nodejs/node">Repository</a> <https://github.com/nodejs/node>
<b>quickjs</b>	Fabrice Bellard / QuickJS	QuickJS is a small and embeddable Javascript engine. <a href="https://bellard.org/quickjs/">Website</a> <https://bellard.org/quickjs/> <a href="https://github.com/bellard/quickjs">Repository</a> <https://github.com/bellard/quickjs>
<b>quickjs-ng</b>	QuickJS-NG / QuickJS-NG	A fork of the original QuickJS project by Fabrice Bellard and Charlie Gordon. <a href="https://quickjs-ng.github.io/quickjs/">Website</a> <https://quickjs-ng.github.io/quickjs/> <a href="https://github.com/quickjs-ng/quickjs">Repository</a> <https://github.com/quickjs-ng/quickjs>
<b>pythonmonkey</b>	Distributive / PythonMonkey	Mozilla JS engine embedded into a Python host environment with cross-platform datatypes and event loop. <a href="https://pythonmonkey.io">Website</a> <https://pythonmonkey.io> <a href="https://github.com/Distributive-Network/PythonMonkey">Repository</a> <https://github.com/Distributive-Network/PythonMonkey>
<b>react-native</b>	Meta / React Native	A framework for building native apps using React. This key represents supported React Native JS runtimes on native platforms (excludes react-native-web). <a href="https://reactnative.dev/">Website</a> <https://reactnative.dev/> <a href="https://github.com/facebook/react-native">Repository</a> <https://github.com/facebook/react-native>
<b>react-server</b>	React / Server Components	Used by React Server Components. <a href="https://react.dev/">Website</a> <https://react.dev/> <a href="https://github.com/facebook/react">Repository</a> <https://github.com/facebook/react>
<b>rhino</b>	Mozilla / Rhino	Rhino is an open-source implementation of JavaScript written entirely in Java. <a href="https://rhino.github.io/">Website</a> <https://rhino.github.io/> <a href="https://github.com/mozilla/rhino">Repository</a> <https://github.com/mozilla/rhino>
<b>wasmer</b>	Wasmer / Wasmer Edge	The JavaScript runtime that brings JavaScript to Wasmer Edge. <a href="https://wasmer.io/products/edge">Website</a> <https://wasmer.io/products/edge> <a href="https://github.com/wasmerio/winterjs">Repository</a> <https://github.com/wasmerio/winterjs>
<b>workerd</b>	Cloudflare / workerd	The JavaScript / WebAssembly runtime that powers Cloudflare Workers. <a href="https://workers.cloudflare.com/">Website</a> <https://workers.cloudflare.com/> <a href="https://github.com/cloudflare/workerd">Repository</a> <https://github.com/cloudflare/workerd>

## 5 Governance and modification process

### 5.1 Proposing new runtime key entries

All [ECMAScript runtimes](#) are welcome to propose a key for inclusion in this Report, subject to the structure as described in [3](#). Runtimes which contribute to this Report are encouraged, but not required, to participate in Ecma TC55.

#### 5.1.1 Proposal process

1. The proposer creates a pull request in the [WinterTC55/runtime-keys GitHub repository](https://github.com/WinterTC55/runtime-keys) <<https://github.com/WinterTC55/runtime-keys>>.
2. The pull request adds the runtime entry to the machine-readable source data file (**runtime-keys.json**).
3. The entry includes all mandatory metadata fields as described in [3.5](#).
4. The entry is inserted in alphabetical order by key.
5. The pull request is reviewed by TC55 participants.
6. At least two delegates within TC55 not directly affiliated with the proposed runtime approve the pull request.
7. The pull request is formally approved at a TC55 plenary meeting before merging.
8. After plenary approval, designated editors merge the pull request and update the current draft Report.

#### 5.1.2 Expert review

All proposed entries undergo expert review to ensure compliance with the naming guidelines specified in [3.3](#) and other expectations. Reviewers evaluate whether:

- the proposed key is appropriately specific and non-generic,
- the runtime has demonstrated proof of use or adoption,
- the key does not conflict with existing entries or Browserslist identifiers, and
- the metadata is complete and accurate.

TC55 may reject a [runtime key](#) submission for any reason, including those not enumerated in the above sections. In such a scenario, the Committee will provide that reason and information on how to resolve any issue.

### 5.2 Modifying entries

#### 5.2.1 Metadata modification

Modifications to runtime metadata (organization, name, description, website, or repository) follow the same process as adding entries:

1. The proposer creates a pull request modifying the entry in **runtime-keys.json**.
2. The proposal continues as described starting at step [5](#) in the approval process.

#### 5.2.2 Key immutability

To prevent breaking existing tools and configurations that depend on [runtime keys](#), **runtime keys are immutable** and will not be modified except in extraordinary circumstances requiring plenary consensus.

#### 5.2.3 Deprecation

[Runtime keys](#) may be deprecated to indicate that a runtime project is inactive or discontinued. To deprecate a key:

1. The proposer creates a pull request setting the **deprecated** field to **true** in **runtime-keys.json**.
2. The proposal continues as described starting at step [5](#) in the approval process.

Deprecated keys may be undeprecated if the original runtime project resumes activity, following the same approval process. Implementations should not add support for deprecated keys.

**NOTE** Deprecated keys remain in the Report to preserve historical information and prevent reuse that could cause confusion.

### 5.3 Publication cadence

This Report is published on a semi-annual basis as an ECMA Technical Report at <https://runtime-keys.proposal.wintercg.org/>. Changes approved at TC55 plenary meetings between publication cycles can be found in the publicly-available draft maintained in the WinterTC runtime-keys repository.

## Annex A (normative)

### Machine-readable canonical source

The authoritative, machine-readable form of the [runtime key](#) source is maintained as a JSON file: [runtime-keys.json](#) <<https://github.com/WinterTC55/runtime-keys/blob/main/runtime-keys.json>>.

#### A.1 JSON schema

The [runtime keys](#) source data JSON file conforms to the following structure:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Runtime Keys",
  "description": "A collection of runtime keys for ECMAScript runtime environments",
  "version": "1.0.0",
  "lastModified": "YYYY-MM-DD",
  "runtimes": [
    {
      "organization": "string (required)",
      "name": "string (required)",
      "key": "string (required)",
      "description": "string (required)",
      "website": "string | null (optional)",
      "repository": "string | null (optional)",
      "deprecated": "boolean (required, default: false)"
    }
  ]
}
```

#### A.2 Accessing the source data

The current machine-readable source is available at:

- Draft version:  
<https://github.com/WinterTC55/runtime-keys/blob/main/runtime-keys.json>
- Published version: To be determined upon Ecma TR publication



## Annex B (informative)

### Example usage

This Annex provides informative examples of how [runtime keys](#) may be used in practice.

NOTE This specification does not prescribe how [runtime keys](#) should be used. The following examples are illustrative only.

#### B.1 package.json example

A library package might use [runtime keys](#) in its `package.json` to specify conditional exports and runtime version requirements:

```
{
  "name": "example-library",
  "version": "1.0.0",
  "exports": {
    "node": "./dist/node/index.js",
    "deno": "./dist/deno/index.js",
    "bun": "./dist/bun/index.js",
    "workerd": "./dist/workerd/index.js"
  },
  "engines": {
    "node": ">=18",
    "deno": ">=1.30",
    "bun": ">=1.0"
  }
}
```



## B.2 Configuration File Example

A build tool or framework might use [runtime keys](#) to specify runtime-specific configuration:

```
{
  "runtimes": {
    "node": {
      "outputDir": "./dist/node",
      "target": "ES2022"
    },
    "deno": {
      "outputDir": "./dist/deno",
      "target": "ES2022"
    },
    "workerd": {
      "outputDir": "./dist/workerd",
      "target": "ES2022",
      "polyfills": ["node:buffer"]
    }
  }
}
```

## Bibliography

ECMA-262, *ECMAScript® Language Specification*  
<https://tc39.es/ecma262/multipage/>



## Software License

Ecma International  
Rue du Rhone 114  
CH-1204 Geneva  
Tel: +41 22 849 6000  
Fax: +41 22 849 6001  
Web: <https://ecma-international.org/>

All Software contained in this document ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT <https://ecma-international.org/memento/codeofconduct.htm> FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

