

DISPATCH
Internet-Draft
Updates: [4329](#) (if approved)
Intended status: Informational
Expires: May 3, 2020

M. Borins
M. Bynens
Google
M. Miller
Mozilla
B. Farias
October 31, 2019

**ECMAScript Media Types Updates
draft-ietf-dispatch-javascript-mjs-05**

Abstract

This document proposes updates to the ECMAScript media types, superseding the existing registrations for "application/javascript" and "text/javascript" by adding an additional extension and removing usage warnings. This document updates [RFC4329](#), "Scripting Media Types".

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Background	2
3.	Security Considerations	3
4.	IANA Considerations	3
4.1.	Common Javascript Media Types	4
4.1.1.	text/javascript	4
4.2.	Historic Javascript Media Types	5
4.2.1.	application/ecmascript	5
4.2.2.	application/javascript	6
4.2.3.	application/x-ecmascript	7
4.2.4.	application/x-javascript	8
4.2.5.	text/ecmascript	9
4.2.6.	text/javascript1.0	10
4.2.7.	text/javascript1.1	11
4.2.8.	text/javascript1.2	12
4.2.9.	text/javascript1.3	13
4.2.10.	text/javascript1.4	14
4.2.11.	text/javascript1.5	15
4.2.12.	text/jscript	16
4.2.13.	text/livescript	17
4.2.14.	text/x-ecmascript	18
4.2.15.	text/x-javascript	19
5.	References	20
5.1.	Normative References	20
5.2.	Informative References	20
Appendix A.	Acknowledgements	20
	Authors' Addresses	20

[1.](#) Introduction

This document updates the existing media types for the ECMAScript programming language. It supersedes the media types registrations in [\[RFC4329\]](#) for "application/javascript" and "text/javascript".

[2.](#) Background

In order to formalize support for modular programs, [\[ECMA-262\]](#) (starting with 6th Edition) defines two top-level goal symbols (or roots to the abstract syntax tree) for the ECMAScript grammar: Module and Script. The Script goal represents the more stand-alone structure where the code executes in the global scope, while the

Module goal represents the module system built into ECMAScript starting with 6th Edition.

This separation means that (in the absence of additional information) there are two possible interpretations for any given ECMAScript Source Text. The TC39 standards body for ECMAScript has determined that media types are outside of their scope of work [[TC39-MIME-ISSUE](#)].

It is not possible to fully determine if a Source Text of ECMAScript is meant to be parsed in the Module or Script grammar goals based upon content alone. Therefore, scripting environments must use out of band information in order to determine what goal a Source Text should be treated as. To this end some scripting environments have chosen to adopt a new file extension of .mjs for determining the goal of a given Source Text.

3. Security Considerations

Module scripts in ECMAScript can request the fetching and processing of additional scripts, called importing. Implementations that support modules need to ensure these scripts are processed the same as scripts processed directly. Further, there may be additional privacy and security concerns depending on the location(s) the original script and its imported modules are obtained from. For instance, a script obtained from "host-a.example" could request to import a script from "host-b.example", which could expose information about the executing environment (e.g., IP address) to "host-b.example".

With the addition of SharedArrayBuffer objects in ECMAScript version 8, it may be possible to implement a high-resolution timer which could lead to certain types of timing and side-channel attacks (e.g., [[SPECTRE](#)]). Implementations may wish to take steps to mitigate this concern, such as disabling or removing support for SharedArrayBuffer objects, or take additional steps to ensure access to this shared memory is only accessible between execution contexts that have some form of mutual trust.

All other security considerations from [[RFC4329](#)] still apply.

4. IANA Considerations

The media type registrations herein are divided into two major categories: the sole media type "text/javascript" which is now in common usage, and all of the media types that are obsolete.

For both categories, The ECMAScript media types are to be updated to point to a non-vendor specific standard undated specification of ECMAScript. In addition, a new file extension of .mjs is to be added to the list of file extensions with the restriction that it must correspond to the Module grammar of [\[ECMA-262\]](#). Finally, the [\[HTML\]](#) specification uses "text/javascript" as the default media type of ECMAScript when preparing script tags; therefore, "text/javascript" intended usage is to be moved from OBSOLETE to COMMON.

[4.1.](#) Common Javascript Media Types

[4.1.1.](#) text/javascript

Type name: text

Subtype name: javascript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: COMMON

Restrictions on usage: The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.](#) Historic Javascript Media Types

The following media types are added or updated for historical purposes. All herein have an intended usage of OBSOLETE, and are not expected to be in use with modern implementations.

[4.2.1.](#) application/ecmascript

Type name: application

Subtype name: ecmascript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .es, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.2.](#) application/javascript

Type name: application

Subtype name: javascript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>.

[4.2.3.](#) application/x-ecmascript

Type name: application

Subtype name: x-ecmascript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .es, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.4.](#) application/x-javascript

Type name: application

Subtype name: x-javascript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.5.](#) text/ecmascript

Type name: text

Subtype name: ecmascript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .es, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.6.](#) text/javascript1.0

Type name: text

Subtype name: javascript1.0

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.7.](#) text/javascript1.1

Type name: text

Subtype name: javascript1.1

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.8.](#) text/javascript1.2

Type name: text

Subtype name: javascript1.2

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.9.](#) text/javascript1.3

Type name: text

Subtype name: javascript1.3

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.10](#). text/javascript1.4

Type name: text

Subtype name: javascript1.4

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.11.](#) text/javascript1.5

Type name: text

Subtype name: javascript1.5

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: This media type is obsolete; current implementations should use text/javascript as the only JavaScript/ECMAScript media type. The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.12.](#) text/jscript

Type name: text

Subtype name: jscript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.13. text/livescript](#)

Type name: text

Subtype name: livescript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.14.](#) text/x-ecmascript

Type name: text

Subtype name: x-ecmascript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .es, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [\[ECMA-262\]](#).

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

[4.2.15.](#) text/x-javascript

Type name: text

Subtype name: x-javascript

Required parameters: none

Optional parameters: charset, see [section 4.1 of \[RFC4329\]](#).

Encoding considerations: Encoding is host dependent with differences in byte order marks, the charset parameter, and text preprocessing.

Security considerations: See [section 5 of \[RFC4329\]](#).

Interoperability considerations: See notes in various sections of [\[RFC4329\]](#).

Published specification: [\[\[RFCXXXX\]\]](#)

Applications which use this media type: Script interpreters as discussed in [\[RFC4329\]](#).

Additional information:

Magic number(s): n/a

File extension(s): .js, .mjs

Macintosh File Type Code(s): TEXT

Person & email address to contact for further information: See Author's Address section.

Intended usage: OBSOLETE

Restrictions on usage: The .mjs file extension signals that the file represents a JavaScript module. Execution environments that rely

on file extensions to determine how to process inputs parse .mjs files using the Module grammar of [ECMA-262].

Author: See Author's Address section.

Change controller: IESG <iesg@ietf.org>

5. References

5.1. Normative References

[ECMA-262]

Ecma International, "Standard ECMA-262: ECMAScript Language Specification", August 2017, <<https://ecma-international.org/publications/standards/Ecma-262.htm>>.

[RFC4329] Hoehrmann, B., "Scripting Media Types", [RFC 4329](#), DOI 10.17487/RFC4329, April 2006, <<https://www.rfc-editor.org/info/rfc4329>>.

5.2. Informative References

[HTML] WHATWG, "HTML Living Standard", August 2017, <<https://html.spec.whatwg.org/multipage/scripting.html#prepare-a-script>>.

[SPECTRE] Kocher, P., Fogh, A., Gerkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., and Y. Yarom, "Spectre Attacks: Exploiting Speculative Execution", January 2018, <<https://arxiv.org/abs/1801.01203>>.

[TC39-MIME-ISSUE]

TC39, "Add `application/javascript+module` mime to remove ambiguity", August 2017, <<https://web.archive.org/web/20170814193912/https://github.com/tc39/ecma262/issues/322>>.

Appendix A. Acknowledgements

The authors would like to thank Suresh Krishnan, Alexey Melnikov, Mark Nottingham, James Snell, Adam Roach, and Allen Wirfs-Brock for their guidance throughout this process.

Authors' Addresses

Myles Borins
Google

Email: mylesborins@google.com

Mathias Bynens
Google

Email: mths@google.com

Matthew A. Miller
Mozilla

Email: linuxwolf+ietf@outer-planes.net

Bradley Farias

Email: bradley.meck@gmail.com

