

Workgroup: Network File System Version 4  
Internet-Draft:  
draft-rmacklem-nfsv4-new-attributes-01  
Published: 15 January 2022  
Intended Status: Standards Track  
Expires: 19 July 2022  
Authors: R. Macklem  
FreeBSD

## **New Attributes for Network File System Version 4, Minor Version 2**

### **Abstract**

This document proposes several new recommended attributes that extend the Network File System Version 4, Minor Version 2 protocol (NFSv4.2). All of these new attributes would be read-only, per file system attributes.

### **Note**

This note is to be removed before publishing as an RFC.

Discussion of this draft occurs on the [NFSv4 working group mailing list](https://mailarchive.ietf.org/arch/browse/nfsv4/), archived at <https://mailarchive.ietf.org/arch/browse/nfsv4/>. Working Group information is available at <https://datatracker.ietf.org/wg/nfsv4/about/>.

### **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 19 July 2022.

### **Copyright Notice**

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

- [1. Introduction](#)
- [2. Requirements Language](#)
- [3. Protocol Extension Considerations](#)
- [4. OPTIONAL New Attributes - List and Definition References](#)
  - [4.1. Definitions of new recommended attributes](#)
    - [4.1.1. Attribute 83: supported\\_ops](#)
    - [4.1.2. Attribute 84: dir\\_cookie\\_rising](#)
    - [4.1.3. Attribute 85: seek\\_granularity](#)
    - [4.1.4. Attribute 86: mandatory\\_br\\_locks](#)
    - [4.1.5. Attribute 87: max\\_xattr\\_len](#)
- [5. Implementation Status](#)
  - [5.1. FreeBSD NFS server and client](#)
- [6. References](#)
  - [6.1. Normative References](#)
  - [6.2. Informational References](#)
- [Acknowledgments](#)
- [Author's Address](#)

## 1. Introduction

Implementation experience with NFSv4.2 has identified that some additional attributes providing per file system information to clients are useful. This document identifies an important set of additional recommended attributes.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

## 3. Protocol Extension Considerations

This document presents an extension to minor version 2 of the NFSv4 protocol as described in [[RFC8178](#)]. It describes new OPTIONAL features. NFSv4.2 servers and clients implemented without knowledge

of this extension will continue to interoperate with clients and servers that are aware of the extension (whether or not they support it).

Note that [\[RFC7862\]](#) does not define NFSv4.2 as non-extensible, so [\[RFC8178\]](#) treats it as an extensible minor version. This Standards Track RFC extends NFSv4.2 but does not update [\[RFC7862\]](#) or [\[RFC8178\]](#).

#### 4. OPTIONAL New Attributes - List and Definition References

The list of New **OPTIONAL** attributes appears in [Table 1](#). The meaning of the columns of the table are:

**Name:** The name of the attribute.

**Id:** The number assigned to the attribute.

**Data Type:** The XDR data type of the attribute.

**Acc:** Access allowed to the attribute. R means read-only.

**Defined in:** The section of this specification that describes the attribute.

Name	Id	Data Type	Acc	Defined in:
supported_ops	83	bitmap4	R	<a href="#">Section 4.1.1</a>
dir_cookie_rising	84	bool	R	<a href="#">Section 4.1.2</a>
seek_granularity	85	uint64_t	R	<a href="#">Section 4.1.3</a>
mandatory_br_locks	86	bool	R	<a href="#">Section 4.1.4</a>
max_xattr_len	87	uint64_t	R	<a href="#">Section 4.1.5</a>

Table 1

#### 4.1. Definitions of new recommended attributes

##### 4.1.1. Attribute 83: supported\_ops

This bit vector indicates which operations are supported for objects (of the appropriate type) with an fsid matching that of the specified object.

The bit vector is a counted array of 32-bit integers used to contain bit values. The position of the integer in the array that contains the bit corresponding to operation  $n$  can be computed from the expression  $(n / 32)$ , and its bit within that integer is  $(n \bmod 32)$ , where  $n$  is the operation number.

#### **4.1.1.1. Rationale**

Without this attribute, an NFSv4.2 client must attempt an optional operation to determine if the server supports it. This attribute allows the NFSv4.2 client to avoid attempting an optional operation when it is not supported for the file object's file system on the server.

This attribute is likely to be particularly helpful in dealing with OPTIONAL attributes whose support is likely to be different for different file systems.

#### **4.1.2. Attribute 84: dir\_cookie\_rising**

TRUE, if performing the REaddir operation on directories with a matching fsid always returns monotonically increasing directory offset cookies. This includes, if named attributes are supported, directories returned by OPENATTR.

##### **4.1.2.1. Rationale**

If the NFSv4.2 client knows that directory offset cookies in REaddir replies are monotonically increasing, it might make it feasible to implement client side support for directory delegation. As an example, implementing client side directory delegation support is much easier when the directory offset cookies are monotonically increasing, so they can be used to order directory entries. Further, the client needs to know that the directory offset cookies are monotonically increasing before reading a directory, so that acquisition and use of a directory delegation may be done. Client side directory caching may also benefit from monotonically increasing directory offset cookies. For example, the Linux NFSv4.2 client uses a different caching algorithm when directory offset cookies are monotonically increasing.

#### **4.1.3. Attribute 85: seek\_granularity**

This attribute indicates the granularity of unallocated regions for data objects with an fsid matching that of the specified object. Data objects include regular files and, when named attributes are supported, named attributes.

0, if the SEEK operation will not return unallocated regions (holes).

1, if the SEEK operation will return unallocated regions (holes), but of no fixed granularity

> 1, if the SEEK operation will return unallocated regions (holes), which are an exact multiple of this attribute in length.

If this attribute is supported for a file system that does not support the SEEK operation, a value of 0 MUST be returned.

#### **4.1.3.1. Rationale**

A NFSv4.2 client can avoid performing RPCs doing the SEEK operation when this attribute is equal 0 or whenever the client can determine that holes greater or equal to the value of this attribute do not exist in the file.

#### **4.1.4. Attribute 86: mandatory\_br\_locks**

TRUE, if byte range locks obtained on data objects with an fsid matching that of the specified object have mandatory semantics potentially affecting IO operations done on overlapping areas. Data objects include regular files and, when named attributes are supported, named attributes.

##### **4.1.4.1. Rationale**

Applications that work with advisory byte range locks will fail with mandatory byte range locks and vice versa. Given that both forms are allowed yet incompatible, it is necessary to provide a way, other than trial-and-error, to determine which form is supported.

Further, if a client is doing read/write caching of data blocks using a write back caching policy, the caching requires locking if mandatory byte range locking is being enforced by the server. For this case, the client needs to acquire an exclusive byte range lock for each byte range being cached. If it does not do so, a write-back may fail, due to a conflicting lock having been acquired by a different client. Therefore, a client doing this kind of data block caching needs to know if the server is implementing mandatory byte range locking.

#### **4.1.5. Attribute 87: max\_xattr\_len**

The maximum length, in bytes, of the extended attribute that can be set by the SETXATTR operation for file system objects with an fsid matching that of the specified object. The SETXATTR operation is described in the [\[RFC8276\]](#) extension to NFSv4.2.

If this attribute is supported for a file system that does not support the SETXATTR operation, a value of 0 MUST be returned.

##### **4.1.5.1. Rationale**

This attribute allows a NFSv4.2 client to avoid attempting a SETXATTR operation when the length of the extended attribute is greater than the maximum specified by this attribute.

## 5. Implementation Status

This section is to be removed before publishing as an RFC.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs.

Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

### 5.1. FreeBSD NFS server and client

**Organization:** FreeBSD Project

**URL:** <https://www.freebsd.org>

**Maturity:** Prototype software based on the current document.

**Coverage:** The bulk of this specification is implemented.

**Licensing:** BSD

**Implementation experience:** The implementation of these attributes have allowed the NFSv4.2 client to avoid unnecessary RPCs against the server. The current client implementation does not include support for directory delegations nor makes use of the `dir_cookie_rising` attribute. The current client implementation does not support mandatory byte range locking.

## 6. References

### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7862] Haynes, T., "Network File System (NFS) Version 4 Minor Version 2 Protocol", RFC 7862, DOI 10.17487/RFC7862, November 2016, <<https://www.rfc-editor.org/info/rfc7862>>.

**[RFC8174]**

Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

**[RFC8178]**

Noveck, D., "Rules for NFSv4 Extensions and Minor Versions", RFC 8178, DOI 10.17487/RFC8178, July 2017, <<https://www.rfc-editor.org/info/rfc8178>>.

**[RFC8276]**

Naik, M. and M. Eshel, "File System Extended Attributes in NFSv4", RFC 8276, DOI 10.17487/RFC8276, December 2017, <<https://www.rfc-editor.org/info/rfc8276>>.

## **6.2. Informational References**

**[RFC8881]**

Noveck, D., Ed. and C. Lever, "Network File System (NFS) Version 4 Minor Version 1 Protocol", RFC 8881, DOI 10.17487/RFC8881, August 2020, <<https://www.rfc-editor.org/info/rfc8881>>.

## **Acknowledgments**

Thanks go to David Noveck for his suggestions for improving the draft.

## **Author's Address**

Rick Macklem  
FreeBSD Project  
Canada

Email: [rmacklem@uoguelph.ca](mailto:rmacklem@uoguelph.ca)