

N/A	A.P.D. Deason
Internet-Draft	Sine Nomine
Intended status: Informational	March 30, 2011
Expires: October 01, 2011	

Adding a Flexible GetSize RPC Variant to the AFS-3 Volume Service  
draft-deason-afs3-getsizev2-03

## [Abstract](#)

This document describes an additional Rx remote procedure call that may be used to obtain the size of an AFS-3 volume dump from an AFS-3 Volume Service. While an RPC already exists for this purpose (AFSVolGetSize), this document proposes a more flexible mechanism that is able to more accurately determine the size of a volume dump in a greater number of cases.

## **Internet Draft Comments**

Comments regarding this draft are solicited. Please include the AFS-3 protocol standardization mailing list (afs3-standardization@openafs.org) as a recipient of any comments.

## [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 <http://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 October 01, 2011.

## [Copyright Notice](#)

Copyright (c) 2011 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 (<http://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.

## [Table of Contents](#)

- \*1. [Introduction](#)

- \*2. [Conventions Used in this Document](#)
- \*3. [Error Codes](#)
- \*4. [Extant RPC Interface](#)
  - \*4.1. [AFSVolDumpV2](#)
  - \*4.2. [AFSVolGetSize](#)
  - \*4.3. [AFSVolDumpV2 Flags](#)
- \*5. [RPC Interface](#)
  - \*5.1. [AFSVolGetSizeV2](#)
  - \*5.2. [AFSVolGetSizeV2 Flags](#)
- \*6. [Security Considerations](#)
- \*7. [IANA Considerations](#)
- \*8. [AFS-3 Registry Considerations](#)
  - \*8.1. [RPC code points](#)
  - \*8.2. [AFSVolGetSizeV2 flags namespace](#)
  - \*8.3. [AFSVolDumpV2 flags namespace](#)
- \*9. [Acknowledgments](#)
- \*10. [References](#)
  - \*10.1. [Normative References](#)
  - \*10.2. [Informative References](#)
- \*[Author's Address](#)

## **1. Introduction**

AFS-3 provides a service (the "Volume Service") [\[AFS3-VVL\]](#) which allows administrators to examine and manage AFS-3 volumes via several Rx remote procedure calls (RPCs) in the AFSVol package. One of these RPCs, AFSVolDump, can be used to obtain a dump of an AFS-3 volume that can be stored on disk, usually for backup or archival purposes. Another RPC, AFSVolDumpV2, was later added to allow the caller to make slight modifications to what data is contained in the dump by way of specifying certain flags in an additional argument.

To allow computation of the size of volumes returned by the AFSVolDump RPC, an AFSVolGetSize RPC was provided. However, no analogous RPC exists to determine what the size of a volume dump obtained via AFSVolDumpV2 would be if any flags were specified in the AFSVolDumpV2 call. This document describes a new RPC, AFSVolGetSizeV2, to fill this gap, to allow computation of the sizes of volume dumps generated by AFSVolDumpV2.

## **2. Conventions Used in this Document**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

## **3. Error Codes**

The existing VOLSERBAD\_ACCESS, VOLSERTRELE\_ERROR, and ENOENT error codes are used by the new RPC defined within this document. The meaning and value of these error codes can be found in [\[AFS3-VVL\]](#), with the exception of ENOENT's value. ENOENT is defined to have the value 2.

## **4. Extant RPC Interface**

The newly-defined RPC AFSVolGetSizeV2 has very similar behavior to the RPCs AFSVolDumpV2 and AFSVolGetSize. While these RPCs already exist and are in use, there is no existing document describing their behavior in significant detail. Their behavior is briefly described here, so the role of AFSVolGetSizeV2 may be better understood.

### **4.1. AFSVolDumpV2**

The AFSVolDumpV2 RPC generates a canonical dump of the contents of a volume associated with a volume transaction. It is identical to the AFSVolDump RPC [\[AFS3-VVL\]](#), except for the addition of the flags parameter.

```
AFSVolDumpV2(IN afs_int32 fromTrans, IN afs_int32 fromDate,
             IN afs_int32 flags) split = 65538;
```

Input arguments:

#### **fromTrans**

A transaction ID as returned by the RPC AFSVolTransCreate.

#### **fromDate**

If 0, the generated volume dump will be a "full dump", and will contain all data in the volume. Otherise, this represents a calendar date, represented as the number of seconds since the Unix Epoch. The

generated dump will then only contain files that have been modified on or after the specified calendar date.

## flags

Zero or more flags combined via a bitwise OR operation. The available flags are described below in [Section 4.3](#).

As with AFSVolDump, the generated volume dump for AFSVolDumpV2 appears in the Rx stream for the split call [\[AFS3-RX\]](#) if there are no errors. The call is aborted with an error for all of the same conditions where AFSVolDump aborts with an error as described in [\[AFS3-VVL\]](#), with the appropriate error code.

## 4.2. AFSVolGetSize

The AFSVolGetSize RPC calculates what the size of a volume dump obtained from calling AFSVolDump [\[AFS3-VVL\]](#) would be. This allows dump sizes to be calculated without needing to generate and transfer the entire volume dump.

```
struct volintSize {
    afs_uint64 dump_size;
};
```

```
AFSVolGetSize(IN afs_int32 fromTrans, IN afs_int32 fromDate,
              OUT struct volintSize *size) = 65537;
```

Input arguments:

### fromTrans

A transaction ID as returned by the RPC AFSVolTransCreate.

### fromDate

The meaning and format of this argument is the same as the fromDate argument described in [Section 4.1](#).

Output arguments:

### size

The size of the dump, represented in octets.

The call is aborted with an error for all of the same conditions where AFSVolDump aborts with an error as described in [\[AFS3-VVL\]](#), with the appropriate error code.

### [4.3. AFSVolDumpV2 Flags](#)

The AFSVolDumpV2 RPC allows the caller to optionally specify a set of 0 or more flags which slightly alter the behavior of the AFSVolDumpV2 call. Only one such flag is currently defined:

**VOLDUMPV2\_OMITDIRS = 1**

Normally, an incremental volume dump generated by AFSVolDump or AFSVolDumpV2 includes the data for all directories in the volume, even if nothing in that directory has changed. This is to allow data in an incremental dump to still be easily accessible even if the incremental dump is the only dump available. If the flag VOLDUMPV2\_OMITDIRS is specified, the generated volume dump will only contain directory data if the directory was modified at or after the specified fromDate. This reduces the size of the dump, but can make it more difficult to retrieve data from the dump if the original full volume dump is lost.

## [5. RPC Interface](#)

One new RPC code is defined for the Volume Service: AFSVolGetSizeV2. Additionally, a new namespace for AFSVolGetSizeV2 flags is defined, and one flag in that namespace is defined.

### [5.1. AFSVolGetSizeV2](#)

The AFSVolGetSizeV2 RPC behaves exactly the same way as the existing AFSVolGetSize RPC ([Section 4.2](#)), except for the addition of the flags argument, which behaves similarly to the flags argument of the existing AFSVolDumpV2 RPC ([Section 4.3](#)).

```
AFSVolGetSizeV2(IN afs_int32 fromTrans, IN afs_int32 fromDate,
                IN afs_int32 flags,
                OUT struct volintSize *size) = XXX;
```

Input arguments:

#### **fromTrans**

A transaction ID as returned by the RPC AFSVolTransCreate.

#### **fromDate**

The meaning and format of this argument is the same as the fromDate argument described in [Section 4.1](#). Since this is represented as a signed 32-bit integer, problems may occur if dates beyond 2038 are specified. A future protocol revision will improve the date representation in AFSVol RPCs to address this problem.

## flags

Zero or more flags combined via a bitwise OR operation. The available flags are described below in [Section 5.2](#).

Output arguments:

## size

The size of the dump, represented in octets.

The call is aborted with an error for all of the same conditions where AFSVolDump aborts with an error as described in [\[AFS3-VVL\]](#), with the appropriate error code.

### [5.2. AFSVolGetSizeV2 Flags](#)

The intention of the AFSVolGetSizeV2 RPC is to provide an extension to the AFSVolGetSize RPC, similar to how AFSVolDumpV2 provided an analogous extension to AFSVolDump. As such, the only defined AFSVolGetSizeV2 flag corresponds to the single existing AFSVolDumpV2 flag, though there is no requirement that every AFSVolGetSizeV2 flag have an AFSVolDumpV2 equivalent, or vice versa.

#### **VOLGETSIZEV2\_OMITDIRS = 1**

This is the AFSVolGetSizeV2 analogue of the AFSVolDumpV2 flag VOLDUMPV2\_OMITDIRS. If this flag is set, the volume dump size will be calculated for a dump generated by an AFSVolDumpV2 call where the flag VOLDUMPV2\_OMITDIRS ([Section 4.3](#)) was set. Such a dump does not include directory information for directories that have not changed since the specified fromDate.

## [6. Security Considerations](#)

The results of the AFSVolGetSize and AFSVolGetSizeV2 RPCs are not normally considered sensitive information, though they may be computationally intensive to calculate. As such, current AFS-3 implementations typically use some per-server superuser list (such as the OpenAFS Basic OverSeer UserList) to control access to the AFSVolGetSize RPC to reduce potential Denial of Service attack vectors. While this practice is OPTIONAL for implementations and subject to site-specific policies, any access control restrictions that apply to the AFSVolGetSize RPC SHOULD also apply to AFSVolGetSizeV2 (and vice versa) for consistency.

## [7. IANA Considerations](#)

This document makes no request of the IANA.

## **8. AFS-3 Registry Considerations**

### **8.1. RPC code points**

This document requires the registration of one RPC code point in the AFSVol Rx package for the AFSVolGetSizeV2 RPC detailed above in [Section 5.1](#).

### **8.2. AFSVolGetSizeV2 flags namespace**

This document requests that the AFS-3 Assigned Numbers Registry assumes control over the AFSVolGetSizeV2 flags namespace. AFSVolGetSizeV2 flags have values from 0x1 to 0x80000000, and must be a power of two. This document suggests that the AFS-3 Assigned Numbers Registry assume control of allocating flags in the range 0x00000001-0x00800000, and that flags in the range 0x01000000-0x80000000 are left to Private Use as defined by [\[RFC5226\]](#).

This document defines one initial assignment in the AFSVolGetSizeV2 flags namespace. VOLGETSIZEV2\_OMITDIRS, detailed above in [Section 5.2](#), is assigned the value of 1.

### **8.3. AFSVolDumpV2 flags namespace**

This document requests that the AFS-3 Assigned Numbers Registry assumes control over the AFSVolDumpV2 flags namespace. There is only one such flag currently in use: VOLDUMPV2\_OMITDIRS with the value of 1, as described in [Section 4.3](#). This document suggests an allocation policy for AFSVolDumpV2 flags identical to the allocation policy described in [Section 8.2](#).

## **9. Acknowledgments**

The author thanks David Boyes and Jeffrey T. Hutzelman for their discussion on Rx return codes, and Derrick Brashear for his work and feedback on representing calendar dates in AFSVol.

## **10. References**

### **10.1. Normative References**

<b>[RFC2119]</b>	<a href="#">Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels"</a> , BCP 14, RFC 2119, March 1997.
------------------	---

### **10.2. Informative References**

<b>[RFC5226]</b>	Narten, T. and H. Alvestrand, " <a href="#">Guidelines for Writing an IANA Considerations Section in RFCs</a> ", BCP 26, RFC 5226, May 2008.
<b>[AFS3-RX]</b>	Zayas, E.R., "AFS-3 Programmer's Reference: Specification for the Rx Remote Procedure Call"

	Facility", Transarc Corp. Tech. Rep. FS-00-D164, August 1991.
<b>[AFS3-VVL]</b>	Zayas, E.R., "AFS-3 Programmer's Reference: Volume Server/Volume Location Server Interface", Transarc Corp. Tech. Rep. FS-00-D165, August 1991.

#### [Author's Address](#)

Andrew Deason Deason Sine Nomine Associates 43596 Blacksmith Square  
Ashburn, Virginia 20147-4606 USA Phone: +1 703 723 6673 EMail:  
[adeason@sinenomine.net](mailto:adeason@sinenomine.net)