XAttr Extension for Novell Storage Services™

Developer Kit

August 2015

Novell.

Contents

About This Guide

XAttr Extension for Novell® Storage Services™ (NSS) makes it easy to back up and restore the NSS file metadata that is not currently exposed by other UNIX* functions.

This feature is not supported on the existing NetWare® file system, but works only with the NSS environment that is included in Open Enterprise Server 2015.

This guide contains the following sections:

Audience

This guide is intended for NSS UNIX and Linux* developers who want to read and set the extended metadata attributes for backup and restore purposes.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Documentation Updates

We intend to post this document on the Novell Developer Kit Web site (http://developer.novell.com/ndk) in the near future.

Additional Information

For the related developer support postings for Novell Storage Services, see the entries for File System Services (64-Bit) on the Developer Support Forums (http://developer.novell.com/ndk/devforums.htm) Web site.

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

1 Concepts

Currently, Novell® Storage Services™ (NSS) stores some file metadata in Open Enterprise Server for Linux that is not accessible with common UNIX functions. The Target Service Agent (TSA) for NSS on Linux uses a special message interface to access the additional metadata. However, with the XAttr Extension included in OES 2015, you can now read, back up, and restore this file metadata without converting existing Linux products.

Future NSS releases will remain backward compatible with this feature.

This section covers the following topics:

XAttr Extension presents extra metadata as an uninterpreted data blob to any backup applications. The same data blob can then be passed to setxattr to restore the metadata, while preserving all the backed-up time stamps for the file.

For sample test files that use this functionality, see the `nw_metadata.c`, `libmetadata.c`, and `restore.c` files that are included with this documentation.

1.1 Overview

XAttr Extension uses the common Linux *getxattr* and *setxattr* functions and adds a new *netware.metadata* and *netware.metadata*2 extended attributes to access additional file metadata that is not available by using other command line options.

The *netware.metadata*2 extended attribute has been introduced in OES beginning with OES 2015. It has the zNW_metadata2_s (page 19) structure that supports both Active Directory and eDirectory users.

Basically, you call getxattr to read your file attributes and one of its parameters points to the zNW_metadata_s (page 16) structure that returns the extended metadata. You can then save this data to a backup location. The zNW_metadata_s (page 16) structure supports only eDirectory users.

To restore or set the extended metadata, call setxattr, specifying which fields you want to modify.

By calling these two functions, you can read, modify, back up, and restore all the metadata for your file. For more information on using getxattr and setxattr, see Chapter 2, "Functions," on page 9.

1.2 Data Sizes

Be aware that XAttr is limited to a total data size of 64 KB.

The size of the data returned by the netware.metadata attribute is defined as follows:

Key Word	Size
WORD	u16
LONG	u32
QUAD	u64
SQUAD	s64
GUID_t	u128

1.3 CtimeIsMetadataModTime Command Line Option

Currently, the Linux Semantic Agent (LSA) uses the creation time as the value for ctime. We'd like it to use the modified metadata time as its value, but there might be unknown complications from modifying this value. Instead, we have provided a new NSS command line CtimeIsMetadataModTime option.

This option allows an administrator to select that the ctime value should be the modified metadata time rather than the creation time.

For more information on how to use NSS command line commands, see Using NSS Server Console Commands and Utilities (http://www.novell.com/documentation/oes/nss_enu/data/ajhvfjd.html) in the *Novell Storage Services File System Administration Guide for OES*.

The CtimeIsMetadataModTime option is set in `/opt/novell/nss/conf/nssstart.cfg` and can be set from nsscon, as follows:

```
nss/(no)CtimeIsMetadataModTime
```

2 Functions

The following functions are not supported on the existing NetWare® file system but work only with the backup and restore environment that is included in OES 2015:

getxattr

Reads a file's attributes and metadata.

Syntax

```
ssize_t getxattr (
    const char  *path,
    const char  *name,
    void        *value,
    size_t       size
);
```

Parameters

path

In OES, this parameter is named *file* and points to the name of the file to read attributes from.

name

Specifies the name of the extended attribute that you would like to get. The extended attribute will either be zNW_metadata_s or zNW_metadata2_s.

value

In OES, this parameter is named *xattr* and points to any one of the following structures:

- the zNW_metadata_s (page 16) structure, and it can be used only for eDirectory users.
- the zNW_metadata2_s (page 19) structure, and it can be used for both eDirectory and Active Directory users beginning with OES 2015.

size

In OES, this parameter represents the length of the value that is passed.

listxattr

Lists a file's extended attribute names.

Syntax

```
ssize_t listxattr (
   const char  *path,
   char        *list,
   size_t       size
);
```

Parameters

path

Points to the name of the file to read extended attributes from.

list

Points to the list of extended attributes.

size

Specifies the length of the attribute list.

Remarks

Extended attributes are name value pairs that are associated with files, directories, symbolic links, etc. They are extensions to the normal attributes that are associated with all files in the system.

listxattr retrieves a list of NULL-terminated extended attribute names that are associated with the given path in the filesystem.

listxattr also returns netware.metadata or netware.metadata2 for every file. Utilities can call listxattr to retrieve the metadata necessary for backing up files. However, this is an optional feature which is turned off by default. To turn on this feature, use `nss/ListXattrNWmetadata` in `nsscon` or make it persistent by putting in `nssstart.conf`.

removexattr

Removes the specified extended attribute from a file.

Syntax

```
int removexattr (
    const char  *path,
    const char  *name
);
```

Parameters

path

Points to the name of the file to remove the extended attribute from.

name

Points to the extended attribute to remove.

Return Values

On success, returns zero. On failure, returns -1 and sets errno.

Remarks

Extended attributes are name/value pairs that are associated with files, directories, symbolic links, etc. They are extensions to the normal attributes associated with all files in the system.

An extended attribute name is a simple NULL-terminated string. The name includes a namespace prefix. There might be several namespaces associated with an individual extended attribute.

IMPORTANT: This function does not remove the `netware.metadata` or `netware.metadata2` extended attribute, but it does remove all the trustees and directory space restrictions associated with the file or directory.

---

setxattr

Modifies a file's attributes and metadata.

Syntax

```
ssize_t setxattr (
   const char  *path,
   const char  *name,
   void        *value,
   size_t       size,
   int          flags
);
```

Parameters

path

In OES, this parameter is named *file* and points to the name of the file to modify attributes in.

name

Specifies the name of the extended attribute that you would like to get. The extended attribute will either be zNW_metadata_s or zNW_metadata2_s.

value

In OES, this parameter is named *xattr* and points to any one of the following structures:

- ◆ the zNW_metadata_s (page 16) structure, and it can be used only for eDirectory users.
- ◆ the zNW_metadata2_s (page 19) structure, and it can be used for both eDirectory and Active Directory users beginning with OES 2015.

size

In OES, this parameter represents the length of the value that is passed.

flags

Specifies the flag for modification. It can either be XATTR_CREATE or XATTR_REPLACE.

3 Structures

The following structures work with the backup and restore environment that is included in OES 2015:

zNW_metadata_s

Contains the extended file metadata.

Defined In: public/zXattr.h

Syntax

```
typedef struct zNW_metadata_s {
    WORD            nwm_byteorder;
    WORD            nwm_version;
    LONG            nwm_reserved_1;
    QUAD            nwm_modify_mask;
    QUAD            nwm_file_attributes;
    QUAD            nwm_file_attributes_mask;

    QUAD            nwm_time_created;
    QUAD            nwm_time_archived;
    QUAD            nwm_time_modified;
    QUAD            nwm_time_accessed;

    QUAD            nwm_time_meta_data_modified;
    QUAD            nwm_reserved_2;
    GUID_t          nwm_id_owner;

    GUID_t          nwm_id_archiver;
    GUID_t          nwm_id_modifier;

    GUID_t          nwm_id_metadata_modifier;
    SQUAD           nwm_quota_limit;
    LONG            nwm_inherited_rights_mask;
    LONG            nwm_trustee_num;

    zNW_trstuee_s   nwm_trustee[zMAX_TRUSTEES];
} zNW_metadata_s;
```

Fields

nwm_byteorder

Specifies the byte order of the structure.

nwm_version

Specifies the version of the structure.

nwm_reserved_1

Is reserved. Set to 0.

nwm_modify_mask

Specifies the attributes (and corresponding fields in this structure) to modify. For possible values, see Section 4.2, "Modify Mask Values," on page 26.

nwm_file_attributes

Specifies the NSS file attributes. For possible values, see Section 4.1, "File Attribute Values," on page 25.

nwm_file_attributes_mask

Specifies the NSS file attributes to modify.

nwm_time_created

Specifies the UTC time when the file was created.

nwm_time_archived

Specifies the UTC time when the file was last archived.

nwm_time_modified

Specifies the UTC time when the file was last modified.

nwm_time_accessed

Is no longer used. Currently, it is set to zero and nwm_modify_mask clears the zMOD_ACCESSED_TIME bit.

nwm_time_meta_data_modified

Specifies the UTC time when the file's metadata was last modified.

nwm_reserved_2

Is reserved. Set to 0.

nwm_id_owner

Specifies the ID of the person who currently owns the file.

nwm_id_archiver

Specifies the ID of the person who last archived the file.

nwm_id_modifier

Specifies the ID of the person who last modified the file.

nwm_id_metadata_modifier

Specifies the ID of the person who last modified the file's metadata.

nwm_quota_limit

Specifies the quota limit that is currently assigned to the file.

nwm_inherited_rights_mask

Specifies the inherited rights to return for the file.

nwm_trustee_num

Specifies the number of the file's trustee.

nwm_trustee

Points to the zNW_trustee_s structure, which contains trustee rights and ID information.

zNW_trustee_s

Contains the rights and ID information for trustees.

Defined In: public/zXattr.h

Syntax

```
typedef struct zNW_trustee_s {
    GUID_t   nwt_id;
    QUAD     nwt_rights;
    QUAD     nwt_reserved_2;
} zNW_trustee_s;
```

Fields

nwt_id

Specifies the trustee ID.

nwt_rights

Specifies the rights for the given trustee.

nwt_reserved_2

Is reserved. Set to 0.

Remarks

You can control the total size of zNW_trustee_s by defining zMAX_TRUSTEES before you include the `public/zXattr.h` file.

In order for this structure to fit in the 64 KB size limitation, we picked 2043 bytes as the maximum size of this structure, as the following snippet (from the `public/zXattr.h` file) shows:

```
#ifndef zMAX_TRUSTEES
#define zMAX_TRUSTEES 2043
#endif
```

zNW_metadata2_s

Beginning with OES 2015, the `zNW_metadata2_s` Xattr structure has been added XAttr NDK to support both Active Directory and eDirectory identities.

Defined In: public/zXattr.h

Syntax

```
typedef struct zNW_metadata2_s {
    WORD   nwm_byteorder;
    WORD   nwm_version;
    WORD    nwm_trustee_more;
    WORD   nwm_reserved_1;
    QUAD   nwm_modify_mask;
    QUAD   nwm_file_attributes;
    QUAD   nwm_file_attributes_mask;

    QUAD   nwm_time_created;
    QUAD   nwm_time_archived;
    QUAD   nwm_time_modified;
    QUAD   nwm_time_accessed;

    QUAD   nwm_time_meta_data_modified;
    QUAD   nwm_reserved_2;

    zNW_user_s nwm_id_owner;
    zNW_user_s nwm_id_archiver;
    zNW_user_s nwm_id_modifier;
    zNW_user_s nwm_id_metadata_modifier;

    SQUAD  nwm_quota_limit;
    LONG   nwm_inherited_rights_mask;
    LONG   nwm_trustee_num;

    zNW_trustee2_s nwm_trustee[zMAX_TRUSTEES2];
} zNW_metadata2_s;
```

Fields

nwm_byteorder

Specifies the byte order of the structure.

nwm_version

Specifies the version of the structure.

nwm_trustee_more

This is reserved for future use, and it can be ignored for now.

nwm_reserved_1

Is reserved. Set to 0.

nwm_modify_mask

Specifies the attributes (and corresponding fields in this structure) to modify. For possible values, see Section 4.2, "Modify Mask Values," on page 26.

nwm_file_attributes

Specifies the NSS file attributes. For possible values, see Section 4.1, "File Attribute Values," on page 25.

nwm_file_attributes_mask

Specifies the NSS file attributes to modify.

nwm_time_created

Specifies the UTC time when the file was created.

nwm_time_archived

Specifies the UTC time when the file was last archived.

nwm_time_modified

Specifies the UTC time when the file was last modified.

nwm_time_accessed

Is no longer used. Currently, it is set to zero and nwm_modify_mask clears the zMOD_ACCESSED_TIME bit.

nwm_time_meta_data_modified

Specifies the UTC time when the file's metadata was last modified.

nwm_reserved_2

Is reserved. Set to 0.

nwm_id_owner

Specifies the ID of the person who currently owns the file.

nwm_id_archiver

Specifies the ID of the person who last archived the file.

nwm_id_modifier

Specifies the ID of the person who last modified the file.

nwm_id_metadata_modifier

Specifies the ID of the person who last modified the file's metadata.

nwm_quota_limit

Specifies the quota limit that is currently assigned to the file.

nwm_inherited_rights_mask

Specifies the inherited rights to return for the file.

nwm_trustee_num

Specifies the number of the file's trustee.

nwm_trustee[zMAX_TRUSTEES2]

Points to the zNW_trustee2_s structure, which contains trustee rights and ID information.

Remarks

zMAX_TRUSTEES2 is defined as 740 that signifies the number of trustees that can be accommodated in the zNW_metadata2_s structure.

```
#define zMAX_TRUSTEES2 740
```

zNW_trustee2_s

Beginning with OES 2015, the `zNW_trustee2_s` structure has been added XAttr NDK to support the trustee information for both Active Directory and eDirectory identities.

Defined In: public/zXattr.h

Syntax

```
typedef struct zNW_metadata2_s {
    zNW_user_s nwt_id;
    QUAD nwt_rights;
    QUAD nwt_reserved_2;
} zNW_metadata2_s;
```

Fields

nwt_id

Specifies the trustee ID.

nwt_rights

Specifies the trustee rights.

nwt_reserved_2

Is reserved. Set to 0.

zNW_user_s

Beginning with OES 2015, the `zNW_user_s` structure has been added to the XAttr NDK, and it contains the identity information and its type: Active Directory or eDirectory.

Defined In: public/zXattr.h

Syntax

```
typedef struct zNW_user_s {
    LONG is_aduser;
    zNW_id_u id;
} zNW_user_s;
```

Fields

is_aduser

Specifies whether the user is an Active Directory user or not.

id

Specifies the user ID.

zNW_id_u

Beginning with OES 2015, the `zNW_id_u` union has been added to XAttr NDK, and it contains the actual user identity information (SID for Active Directory and GUID for eDirectory).

Defined In: public/zXattr.h

Syntax

```
typedef union zNW_id_u {
SID_t aduser;
  GUID_t ediruser;
} zNW_id_u;
```

Fields

aduser

> Specifies the Active Directory user's SID.

ediruser

> Specifies the eDirectory user's GUID.

4 Values

The following values are defined for the backup and restore environment that is included in OES 2015:

4.1 File Attribute Values

The following values are defined for the nwm_file_attributes field in the zNW_metadata_s (page 16) structure:

Table 4-1 *File Attribute Values*

Value	Name	Description
0x00000001	zFA_READ_ONLY	Read only file.
0x00000002	zFA_HIDDEN	Hidden file.
0x00000004	zFA_SYSTEM	System file.
0x00000008	zFA_EXECUTE	Executable file.
0x00000010	zFA_SUBDIRECTORY	
0x00000020	zFA_ARCHIVE	Archiveable file.
0x00000080	zFA_SHAREABLE	Shareable file.
0x00000700	zFA_SMODE_BITS	
0x00000800	zFA_NO_SUBALLOC	
0x00001000	zFA_TRANSACTION	
0x00002000	zFA_NOT_VIRTUAL_FILE	Valid only on a volume with the zATTR_VIRTUAL_FILES attribute.
0x00010000	zFA_IMMEDIATE_PURGE	
0x00020000	zFA_RENAME_INHIBIT	
0x00040000	zFA_DELETE_INHIBIT	
0x00080000	zFA_COPY_INHIBIT	
0x00100000	zFA_IS_ADMIN_LINK	The file contains persistent administrator link information.
0x00200000	zFA_IS_LINK	
0x00400000	zFA_REMOTE_DATA_ACCESS	File was migrated.
0x00800000	zFA_REMOTE_DATA_INHIBIT	File won't allow data migration.

Value	Name	Description
0x02000000	zFA_COMPRESS_FILE_IMMEDIATELY	
0x04000000	zFA_DATA_STREAM_IS_COMPRESSED	Set per data stream directory entry.
0x08000000	zFA_DO_NOT_COMPRESS_FILE	
0x20000000	zFA_CANT_COMPRESS_DATA_STREAM	No space is saved by compressing this file.
0x40000000	zFA_ATTR_ARCHIVE	
0x80000000	zFA_VOLATILE	Data is volatile (no oplocks).

4.2 Modify Mask Values

The following values are defined for the nwm_modify_mask field in the structure:

Table 4-2 *Modify Mask Values*

Value	Name	Description
0x00000001	zMOD_FILE_ATTRIBUTES	
0x00000002	zMOD_CREATED_TIME	
0x00000004	zMOD_ARCHIVED_TIME	
0x00000008	zMOD_MODIFIED_TIME	
0x00000010	zMOD_ACCESSED_TIME	This bit is no longer used. Instead, the nwm_time_accessed field is set to zero and this bit is cleared.
0x00000020	zMOD_METADATA_MODIFIED_TIME	
0x00000040	zMOD_OWNER_ID	
0x00000080	zMOD_ARCHIVER_ID	
0x00000100	zMOD_MODIFIER_ID	
0x00000200	zMOD_METADATA_MODIFIER_ID	
0x00000400	zMOD_PRIMARY_NAMESPACE	If the volume is mounted in the LONG namespace, no access is provided to UNIX, Mac, or DOS namespaces. However, because the alternate namespaces are invisible, the generated alternate names are sufficient.
0x00000800	zMOD_DELETED_INFO	
0x00001000	zMOD_MAC_METADATA	
0x00002000	zMOD_UNIX_METADATA	
0x00004000	zMOD_EXTATTR_FLAGS	
0x00008000	zMOD_VOL_ATTRIBUTES	

Value	Name	Description
0x00010000	zMOD_VOL_NDS_OBJECT_ID	
0x00020000	zMOD_VOL_MIN_KEEP_SECONDS	
0x00040000	zMOD_VOL_MAX_KEEP_SECONDS	
0x00080000	zMOD_VOL_LOW_WATER_MARK	
0x00100000	zMOD_VOL_HIGH_WATER_MARK	
0x00200000	zMOD_POOL_ATTRIBUTES	
0x00400000	zMOD_POOL_NDS_OBJECT_ID	
0x00800000	zMOD_VOL_DATA_SHREDDING_COUNT	
0x01000000	zMOD_VOL_QUOTA	
0x02000000	zMOD_DIR_QUOTA	
0x04000000	zMOD_READ_AHEAD_BLOCKS	
0x08000000	zMOD_INH_RIGHTS_MASK	
0x10000000	zMOD_ALL_TRUSTEES	

A Revision History

The following table outlines all the changes that have been made to the XAttr Extension for Novell® Storage Services™ documentation (in reverse chronological order).

March 11, 2015	Added information regarding the new structures (zNW_metadata2_s (page 19), zNW_trustee2_s (page 21), zNW_user_s (page 22), and zNW_id_u (page 23))that have been introduced to support both Active Directory and eDirectory users.
October 11, 2006	Updated zNW_metadata_s (page 16) to show that the nwm_time_accessed field is no longer used. Instead, it is set to zero and nwm_modify_mask clears the zMOD_ACCESSED_TIME bit.
June 21, 2006	Added listxattr (page 11) and removexattr (page 12).
March 1, 2006	Added as a new NDK component.
January 23, 2006	Created as a new document.