



Migrating to Extended Attribute Name and Data Spaces

11-February-2011

Andrew Hanushevsky



Scalla: Structured Cluster Architecture for Low Latency Access
©2003-2011 by the Board of Trustees of the Leland Stanford, Jr., University
All Rights Reserved
Produced under contract DE-AC02-76-SFO0515 with the Department of Energy
This code is available under a BSD-style license allowing minimally restricted use.

1	Introduction	5
1.1	Running In Backward Compatibility Mode.....	6
2	Preparing For Conversion	7
3	The Conversion Tool.....	9
3.1	Recommended Names Conversion Sequence	11
3.2	Recommended Space Conversion Sequence.....	11

1 Introduction

This document describes how to migrate to extended attribute name and data spaces used by version 3.1.0 (and above) of **xrootd** and the **File Residency Manager (FRM)**. In previous versions, **xrootd** created special meta-files¹ in the name and data spaces to track information needed to migrate, purge, and recover files using the **FRM** component. The following files may have been created:

- Special “.**lock**” and “**DIR_LOCK**” meta-files were created in the name space if the space was exported **read/write** with **migrate** or **purge** attributes.
- Special “.**pin**” meta-files were created when the associated data file of pinned on disk.
- Special “.**pfm**” meta-files were created when “**oss.space**” directive or “**oss.cache**” directive *with* the “**xa**” option was specified in the configuration file.

In current **xrootd** releases, the information embodied by these special meta-files is now recorded as file system extended attributes of the associated file. For server configurations that did not require meta-files to be created, you *do not need* to perform a meta-file conversion. However, if you used the “**oss.cache**” directive *without* the “**xa**” option, you may still need to convert your data spaces; as described in “Convert Old-Style Data Spaces”.

Current versions of **xrootd** have the capability of running in backward compatibility mode; though, this is not the default. Running the current version of **xrootd** and **FRM** against old-style name and data spaces does not cause any data loss. However, the following two side-effects occur:

- Files created by previous versions of **xrootd** that used meta-files to track file status cannot be migrated or purged by the **FRM**.
- Files created by current versions of **xrootd** use extended attributes to track file status and cannot be migrated or purged by previous versions of the **FRM**.

¹ Meta-files are files with a special suffix. The filename less the suffix is the name of the data file.

1.1 Running In Backward Compatibility Mode

If you do not have the time to convert one or more servers, you can run those servers in backward compatibility by including the following directive in each configuration file where backward compatibility is needed.

```
oss.runmodeold
```

The **xrootd** and **FRM** daemons will recognize and continue to create meta-files. Please be aware that this is not a long-term solution as backward compatibility mode will eventually be discontinued.

2 Preparing For Conversion

You should answer the following two questions prior to conversion.

Do all file systems used by xrootd support extended attributes?

You can use the “**frm_admin -c *cfile* convert test**” which will tell you if extended attributes are supported for each path mentioned in the configuration file, *cfile*.

All modern file systems support extended attributes, by default. In Linux, however, user extended attributes are normally disabled for **ext2** and **ext3** file systems. If possible, migrate to **ext4** or **xfs** which also perform much better. Otherwise, you will need to turn on extended attributes in the **/etc/fstab** file and remount each file system. This is done by adding the **user_xattr** file system attribute and remounting each file system. For example,

```
#In /etc/fstab
/dev/hda3      /home      ext3  defaults,user_xattr 1 2

#As root
mount -o remount /home
```

Is an old-style data space being used?

If the **xrootd** configuration file contains “**oss.cache**” directives without the “**xa**” option, then you are using an old-style data space. It is highly recommended that you convert to using **xa** data spaces. This is described in “Recommend Space Conversion Sequence” section.

You may have used non-XA data spaces in the past and these are still reachable. In this case, the **frm_admin** command used in the conversion process will indicate whether any old-style data spaces are still reachable and conversion is recommended.

3 The Conversion Tool

The conversion process is handled by the `frm_admin` command. You must run this command on each server that is to be converted, using the *same* username as used by `xrootd`. Below is the command synopsis.

```

frm_admin [options] [cvtparam]

options: [-c cfname ] [-d] [-n name] [-v]

cvtparam: convert [cvtopts] old2new [names] [spaces]

cvtopts: [-a[utoreply]] [-fix]

```

Options

-c *cfname* The configuration file to be used. The default configuration file, `/opt/xrootd/etc/xrootd.cf`, is read if it exists.

-d Turns on debugging.

-h Prints usage information.

-n *name*

Is the name of the server instance for which the command is being performed. See the notes for more information.

-v Prints additional messages where relevant.

Parameters

names

Converts “.lock” and “.pfn” meta-files in the exported name space to extended attributes and removes all known meta-files².

spaces

Converts non-XA³ data spaces to an XA format.

² “DIR_LOCK”, “.mkeep”, “.mlock”, “.mmap”, and “.pin” meta-files.

³ Spaces defined using the `oss.cache` directive without the `xa` option.

Migrating to XA

Cvtops

-autoreply

Automatically replies with the default responses for any prompts that may occur during the conversion.

-fix Attempts to fix problems encountered during the conversion.

Notes

- 1) If you do not specify **names** or **spaces** the default is to convert both **names** and **spaces**.
- 2) You cannot convert systems that are still using “**oss.runmodeold**”.
- 3) The conversion can be run on a live system. This allows you to do a non-disruptive conversion. However, for smoother results we recommend that the **frm_purged** and **frm_xfrd** *not* be running during the conversion. Any **FRM** requests will be automatically queued by the system and resumed once the **FRM** daemons are restarted.

3.1 Recommended Names Conversion Sequence

- 1) Review the configuration file and remove the **oss.runmodeold** directive.
- 2) Kill the **cmsd**, **frm_purged**, **frm_xfrd**, and **xrootd** daemons.
- 3) Restart the **cmsd** and **xrootd** daemons with the new configuration file as they can provide service during the conversions process.
- 4) Minimally, run the following command, using the *same* username as used by **xrootd**, where *cfile* is the new configuration file:
frm_admin -c cfile convert old2new names
- 5) Carefully review the results. For each exported path, you may run
frm_admin -c cfile audit -r names export_path
- 6) If you are satisfied you may wish to continue on to the space conversion, if need be (the **frm_admin convert** procedure will indicate if this is something you should do). Otherwise, restart the **frm_purged** and **frm_xfrd** daemons, as required.

3.2 Recommended Space Conversion Sequence

- 1) Review the configuration file and remove the **oss.runmodeold** directive.
- 2) Kill the **cmsd**, **frm_purged**, **frm_xfrd**, and **xrootd** daemons.
- 3) Restart the **cmsd** and **xrootd** daemons with the new configuration file as they can provide service during the conversions process.
- 4) Minimally, run the following command, using the *same* username as used by **xrootd**, where *cfile* is the new configuration file:
frm_admin -c cfile convert old2new spaces
- 5) Carefully review the results. For each space name, you may run
frm_admin -c cfile audit space spacename
- 6) If you are tracking usage, you should also run
- 7) **frm_admin -c cfile audit -fix usage**
- 8) If you are satisfied, restart the **frm_purged** and **frm_xfrd** daemons, as required.