

# DMF Hints for the Help Desk

A new DMF site's Help Desk may face some unfamiliar questions

Peter Edwards | Sr System Administrator

December 2013

CSIRO IM&T SCIENTIFIC COMPUTING  
[www.csiro.au](http://www.csiro.au)



# Overview

When an organisation starts using DMF, its Help Desk will receive calls on some unfamiliar issues.

This presentation attempts to describe some of the most common ones; they won't necessarily be relevant to all sites.

# Contents

- My terminal session keeps hanging!
- Why did I hit a space quota? Isn't this space meant to be unlimited?
- What's an inode quota and why are you inflicting it on me?
- How can I speed up *rsync*?
- What is "churn" and why do I care?
- The filesystem is too full to hold my files; I'm going somewhere else.
- Why do my files go offline and other people's don't?
- I can't keep my files online.
- From Windows, how can I see which files are online? How can I make them so?
- From MacOS or Linux, how can I see which files are online? How can I make them so?

# My terminal session keeps hanging!

An HSM such as DMF provides near unlimited storage by holding the data of inactive files on backend media such as tape, MAID or slow (SATA) disk which are slower and cheaper than the main high performance disk.

Any attempt to access this data automatically causes the data to be “recalled” to the main disk, which takes some time. While this is happening, the program requiring the data waits for it to become available, at which time it resumes execution.

The length of the delay depends on the nature of the backend media, and how busy it is. This is usually under a second for slow disk and several minutes for tape, though at times of peak demand this could be considerably longer.

(cont'd)

# My terminal session keeps hanging (cont'd)

This effect becomes worse if multiple offline files are accessed, one after another.

To avoid this, it is recommended that the *dmget* command be used to recall multiple files in parallel, as a single batch, before running the program which requires them.

Use of *dmget* is also gentler on the hardware and reduces the impact of one user's work upon that of others.

# Why did I hit a space quota? Isn't this space meant to be unlimited?

DMF manages the disk so that when it starts filling, the largest and/or oldest files have their data automatically written to offline storage (eg: tape) if necessary, and then their online space is released allowing its use for more active data. This takes time, and it's possible for the rate of consumption to outpace DMF's ability to free up space. The disk would then fill and user work would abort.

To prevent this, a per-user space limit is often implemented to stop runaway consumption from endangering other users' work.

If this happens often, the *dmpur* command may be used to volunteer files to go offline sooner than would otherwise happen. A quota increase is not normally the right solution.

# What's an inode quota and why are you inflicting it on me?

DMF is sensitive to large numbers of files/directories, because of the need to scan the entire filesystem at times to collect details of all the files before performing various administrative tasks. Some of these activities are time-critical, for example to prevent the filesystem from filling.

A per-user inode (ie: file or directory) quota is in place to prevent excessive file creation which would cause DMF to become inefficient, wastefully consuming system resources.

If this quota affects you, please consider whether you can use tar, zip or similar to group multiple files into a single file before requesting an increase in quota.

# How can I speed up *rsync*?

- Whether you are *rsync*ing files to or from the DMF-managed disk, you should use the *--whole-file* option to assess whether an existing file needs updating without inspecting its data, which would trigger a *recall* of every file.
- When *rsync*ing files from the DMF-managed disk to elsewhere, it is recommended that you first recall the files with *dmget*.

If you can't predict which files will be transferred, you could do a dry-run *rsync* by adding the *-dry-run* and *--itemize-changes* flags to collect details of files to be transferred, extract the filenames and pipe them into *dmget* in the background. Then start the real *rsync* command, and the *dmget* will probably race ahead.



# What is “churn” and why do I care?

“Churn” is when a new file is migrated by DMF, thereby having multiple copies made in the back-end storage, and deleted/alterd soon after.

Churn has several impacts which we would prefer to avoid:

- The I/O needed to make these copies, which is wasted
- Records in DMF’s databases detailing the data’s location, which are kept for a few months after the file’s deletion/alteration, before being deleted and reused
- The space used in the back-end storage, kept for a couple of months for slow (SATA) disk and potentially for years for MAID and tape based copies

Disk space not managed by DMF is made available for short-term working areas; this should be used instead of the DMF space intended for long-term storage of unaltered files.

Why should **you** care? Because we’ll keep nagging you until you do!

# The filesystem is too full to hold my files; I'm going somewhere else.

DMF manages the disk so that when it starts filling, the largest and/or oldest files have their data automatically written to offline storage (eg: tape) if necessary, and their online space is then released allowing its use for more active data.

So in order to to extract maximum benefit for the fast and expensive disk, DMF is configured to keep the utilisation over 90%.

In other words, don't believe what *df* says about a DMF-managed filesystem! Provided you don't do it too quickly, it will absorb all the data you can throw at it.

And if you do go somewhere else, make sure they do backups!

# Why do my files go offline and other people's don't?

When DMF has to force files offline to prevent a filesystem filling, it assigns a weight to every migratable file in there. This weight is normally based on the space used by the file, and how long it's been since it was last accessed or altered.

It then starts offlining files in order by weight until the filesystem occupancy is below a threshold, chosen by the site but frequently around 80%.

So if your files are larger or used less frequently than the average, then they are more likely to go offline sooner.

# I can't keep my files online.

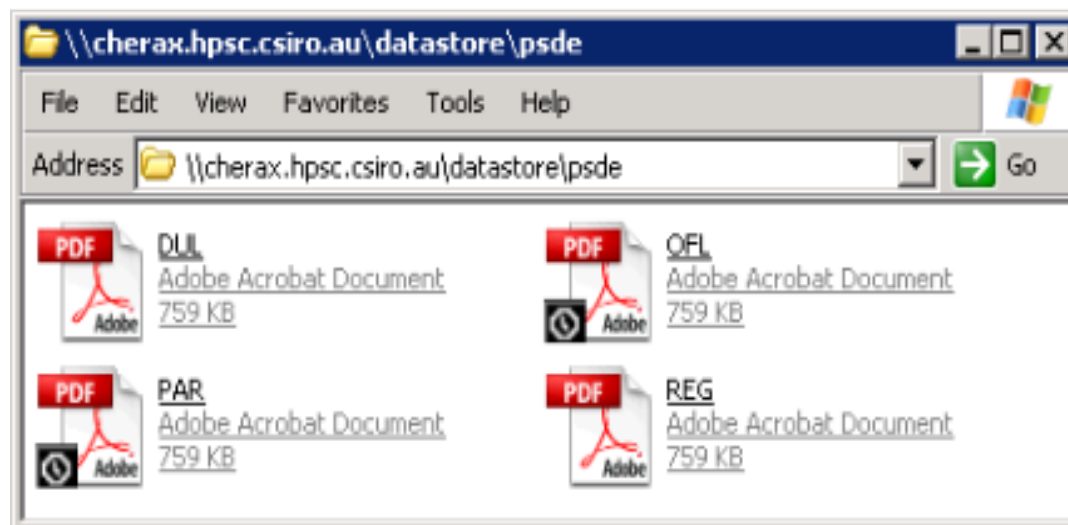
Some users require the presence of several very large files online before processing can start, and sometimes while one file is being recalled to disk another file which was recalled earlier could flip back to offline; a sort of “thrashing” behaviour.

About all the user can do is to specify the “-a” flag on the *dmget* command, which makes freshly recalled files look freshly accessed. This is no guarantee, but it can improve the odds that in such a situation someone else's files will go instead.

The real solution is to get more hardware, probably more high speed disk to enlarge the DMF-managed filesystem.

# From Windows, how can I see which files are online? How can I make them so?

Provided the Samba server has been configured to allow it, Windows Explorer will show offline or partial-state files by modifying their icon to add a clock-face or a letter “X”.



There's currently no good way for Windows users to do a bulk recall of offline files. They really need to run the dmget command on the DMF server or on a CXFS/NFS client of that server somehow.

# From MacOS or Linux, how can I see which files are online? How can I make them so?

If the DMF-managed filesystem is NFS-mounted, SGI provides DMF command-line client utilities (dmls, dmget, etc) packaged as DMG and RPM files. But there is no GUI interface included.

For other protocols (AFP, SMB/CIFS, etc) there is no support at all.

Users of these systems will have to access the DMF server or a CXFS/NFS client of that server somehow.

# Thank you

**CSIRO IM&T Scientific Computing**  
Peter Edwards  
Sr System Administrator

e [peter.edwards@csiro.au](mailto:peter.edwards@csiro.au)  
w [www.hpsc.csiro.au/](http://www.hpsc.csiro.au/)

**CSIRO IM&T SCIENTIFIC COMPUTING**  
[www.csiro.au](http://www.csiro.au)

