Optimal Performance for your MacroView DMF Solution

In this paper we identify 20 points to consider if you want to optimise the performance of your MacroView DMF solution. These points also apply to MacroView Message, the subset of MacroView DMF that runs in Outlook.

1. Avoid having a large number of unique permissions

You should design your SharePoint document store such that you do not have large numbers of breaks in permission inheritance – in other words large numbers of unique permissions.

Designing the structure of the store so that it reflects the structure of your organization often leads to a lot of permission breaks because most users do not have permission to top level nodes (which might correspond to Divisions or Departments) but only to lower-level nodes (correspond to teams or projects). A better approach is to design so that users tend to have access to all areas of a site collection, if they have access at all to that site collection.

The performance impact of having a large number of unique permissions is particularly noticeable in *Browse* mode, and to a lesser extent in *Favorites* mode. It does not impact on *Search* or *Recent* modes.

As it displays the *Browse* mode, MacroView DMF seeks to show all nodes for which the user has permission, even if those nodes are nested below nodes where the user has no permission. To do this MacroView DMF needs to traverse the tree of sub-nodes, checking whether there is at least one sub-node for which the user has permission. Checking sub-nodes is quick if those sub-nodes inherit permissions from their parent node. If not, the traversal can consume significant resources, especially if the tree is large and the user has no permission to any sub-nodes or only to deeply nested nodes.

Filtering improves Performance, especially if Server-side caching is ON

Filtering of sub-nodes makes this process more efficient. This is because DMF will traverse only the nodes that satisfy a filter criterion entered by the user (i.e. title of site / document library title or name of folder / document set contains a nominated string of characters). The filtering will happen automatically if the number of sub-nodes exceeds the *Nodes To Trigger Filter* setting (shipped value 200). The filtering can also be triggered by right-clicking a node and choosing *Filter Site Collections, Filter Sites and Libraries, Filter Folders* as appropriate.

Filtering of sub-sites in a site can be sped up significantly by turning ON MacroView Server-side caching[1]. This is because the information required to perform this filtering is cached in the node being filtered – there is no need to instantiate sub-webs in order to check whether their title satisfies the filter. This instantiation has significant resource overheads.

MacroView server-side caching also provides an option to cache Document Libraries. If this option is ON (see the relevant page in the Administration Console), titles of the document libraries in a site are cached along with the titles of the sub-sites of that site, and filtering of document libraries is quicker.

Ensure at least *Limited* permission to Site and Library nodes

Another technique that can be useful for improving performance in cases where the tree is large and contains a large number of unique permissions is to grant users *Limited* access to all site and library nodes. This does not allow a user to view, open or modify more documents, but it is sufficient to cause MacroView DMF to stop the traverse because it has found a node that the user can access.

[1] Server-side caching is only relevant to implementations using on-premises SharePoint– not available on Office 365.

Folders with Unique Permissions

The *Disable folder count checking*[2] setting can improve performance for designs where there are document libraries containing a large number of folders or document sets with unique permissions (i.e. there is a break in permission inheritance before the folder / document set). Setting *Disable folder count checking* ON will cause MacroView DMF to display the document library with an expansion indicator without first checking whether the user has permission to any of its folders or document sets. This significantly improves the performance of expanding the site that contains the document library. The downside is that the user may find that the expansion indicator on the library is misleading – when he / she expands the library node nothing may be displayed, because there are no folders / document sets which he / she has permission. This will not be a problem in designs where all libraries contain folders.

Performance when expanding a document library node can also be poor when there are a relatively small tree of folders with unique permissions, but with each folder containing a large number of files. In this scenario *Content Type* should NOT be indexed. Indexing *Content Type* causes SharePoint to use the Content Type Iterator, which is appropriate when you need to process all the files, but not when MacroView DMF is expanding a library node (when it needs only to find one item that the user has permission to access).

Other Techniques

There are other ways to obtain good performance when navigating a tree that contains a large number of permission breaks. These include:

- Right-click on a search result in *Search* mode and choose *Open File Location*
- Right-click on a file listed in the *Recent* mode and choose *Open File Location*
- *Search Site Tree* (see point 5 for more details).

2. Use Server-side caching for the web application that contains My Sites

Here we are referring to My Sites in an on-premises SharePoint server – not to the use of One Drive for Business. In an on-premises SharePoint server each My Site is implemented as a separate site collection. In an organization with hundreds of staff there will be hundreds of site collections related to My Sites. Typically any one user will have permission to only some of these My Site site collections.

In such environments the DMF Cache Manager should be enabled. As described above, this will speed up the filtering of site collections, which in turn will make the expansion of top level (web application) nodes much faster.

MacroView DMF applies a further filter to My Sites, so that the only My Site site collections that are displayed to a user are those for his / her colleagues[3]. Note that MacroView DMF applies this further filter AFTER it performs its standard processing of a top-level (web application) node. If the organization has 10,000 staff this means that 10,000 site collections will be processed to check whether their title contains the character string that the user has entered as a filtering criterion, and then some of the site collections resulting from this initial filtering will be discarded because they are not related to colleagues.

A further point to note in relation to My Sites is that you should follow Microsoft's recommended approach of locating My Sites in a separate web application, especially if there is a large number of My Sites. Otherwise the display of other web applications will be made less efficient by the presence of the site collections related to My Sites mixed in with other site collections.

[2] The *Disable folder count checking* setting can be found in the Application Management page for MacroView DMF in SharePoint Central Administration.

[3] Colleagues as specified using the user's own My Site in the SharePoint web browser UA.

3. Use Server-side caching if you are using multiple Content Databases

Many organizations are moving to have large numbers of site collections because this maximizes capacity of the SharePoint document store, as each site collection can potentially be mapped to its own Content Database. MacroView DMF provides best-available support for working with a SharePoint document store that contains a large number of site collections.

While having additional Content Databases certainly enables a significant expansion in capacity of a SharePoint document store it should also be noted that there are resource overheads associated with connecting to each Content Database. The more content databases you have configured, the more important it is to have MacroView server-side caching turned ON.

The reason for this is as described above (see point 1). As you attempt to expand a top-level (web application) node, MacroView needs to check whether you have permission to the site collections that satisfy your filtering criterion. If Server-side caching is ON, the details required to process that filtering are cached in the web application node. This removes the need to instantiate the root sites of every site collection, a process which is made even more expensive by the need to connect to all the content databases being used to store those site collections.

Starting with the MacroView DMF v7.8 December 2014 Update, the performance of the timer jobs that are used to build and maintain the MacroView server-side cache has been dramatically improved in scenarios where multiple content databases are in use. A MacroView DMF customer with 12 content databases reported a reduction in the run time of the Full Cache Update job from approximately 48 hours to less than 5 hours.

4. Avoid retrieving a large number of folder-level default values

Some organizations opt to implement automatic metadata by defining default values at folder level. Internally SharePoint uses a single XML document to store the default values for all folders in a document library. If there are a large number of folders with default values in a library, this XML structure can become quite large.

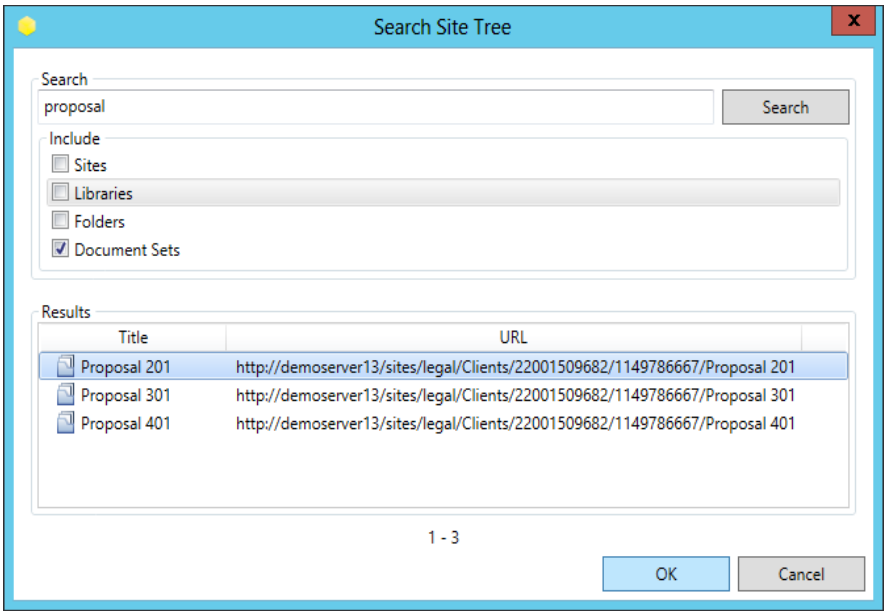
As it saves / uploads a document or email to a folder, the standard action of MacroView DMF is to retrieve the definition of all metadata columns from the server and transfer it to the client PC, where it can be used to display the metadata capture (profiling) dialog. This definition includes any folder-level default values, which means that the large XML structure is retrieved every time a document is saved / uploaded to a folder. This retrieval can consume significant resources and severely impact on performance.

If the design of the DM store is such that there is no need to prompt for metadata, then *Retrieve Folder Level Column Defaults* should be set OFF, which will mean that the XML structure containing defaults for all folders in the document library is not downloaded to the client. This avoids wasting server and bandwidth resources and can significantly speed up saves and uploads.

5. Use *Search Site Tree*

The *Search Site Tree* command of MacroView DMF can be a highly performant way to navigate a SharePoint document store that contains a large number of nodes, particularly if those nodes tend to be uniquely named. *Search Site Tree* uses the SharePoint search engine to perform an indexed search for all nodes whose titles / names contain a nominated string of characters. *Search Site Tree* will then efficiently navigate you to the node that you select in the search results. The overall effect is generally less resource consumption (and better performance) than locating the wanted node by expanding all upper level nodes in the tree.

Newly created nodes will not be found until they have been added to the SharePoint search index by a crawl, but otherwise Search Site Tree is more efficient than filtering, even when Server-side caching is enabled.



Search: proposal

□ Sites
□ Libraries
□ Folders
☑ Document Sets

Title	URL
Proposal 201	http://demoserver13/sites/legal/Clients/22001509682/1149786667/Proposal 201
Proposal 301	http://demoserver13/sites/legal/Clients/22001509682/1149786667/Proposal 301
Proposal 401	http://demoserver13/sites/legal/Clients/22001509682/1149786667/Proposal 401

1 - 3

MacroView DMF v7.8 February 2015 Update and later versions of MacroView DMF allow you to nominate which type(s) of node you are interested in – sites / document libraries / document sets / folders.

This *Search Site Tree* technique is not effective if a lot of nodes have the same title – e.g. every site contains a document library called *Documents*. For this reason a good design tip is to prefix site / library titles and document set / folder names with some meaningful code or number – e.g. Customer Code for customer sites, Project Number for project document sets, etc.

6. Use *Favorites* rather than *Browse* mode

Feedback from many established MacroView DMF customers is that users tend to do most of their interaction with the SharePoint document store from the *Favorites* mode ("*Favorites* mode is my favorite mode"). Using *Favorites* can dramatically reduce the amount of effort related to navigating the document store, which translates to improved user productivity.

Once you navigate to a node in *Browse* mode you can right-click the node and choose *Add to Favorites*. Thereafter navigation to that node is a one-click experience. You can also have the MacroView DMF Push Favorites mechanism customized to create such Favorites for you automatically.

7. Do not suppress *Filtering*

Do not increase the *Nodes to Trigger Filter* value in an attempt to suppress filtering – filtering is a key technique used by MacroView DMF to enable efficient navigation of large SharePoint structures. If you feel that is clumsy to be prompted to filter when you attempt to expand a node that has a large number of sub-nodes, consider using *Favorites* instead or *Browse* as you main mode of operation (see point 6).

8. Do not have a very large *Files in file list* setting

Like any good bandwidth-aware application, MacroView DMF pages the display of files. Before you increase the page size, consider the impact that retrieving hundreds of files will have on bandwidth and performance – very large values of *Files in file list* may lead to timeouts before all files are retrieved. Remember that a file does not have to be displayed in order to be found when you filter on a column.

Note that MacroView DMF will not display search results in *Search* mode if the value of *Files in file list* is above 500.

9. Do not have a large number of Recents

MacroView DMF maintains details of files that you have accessed recently, searches that you have performed recently and areas wherein you have worked recently. You can specify the maximum number of these Recent entries that MacroView DMF should retain (see *Options, General, Recent Items, Maximum Recent Files*).

Having a large number of Recent entries drives up memory usage by the MacroView DMF Local Service and also the volume of data that needs to be written back to you APPDATA area (which decreases performance).

The recommended value for Maximum Recent Files is 30.

10. Do not Increase the List View Threshold

SharePoint 2010 introduced the List View Threshold, which is set at 5,000 for non-administrator users when SharePoint installs. When the user attempts to display a view of a library that requires processing more than 5,000 items, the default action of SharePoint is to cancel the query that it would otherwise send to SQL Server, with the result that the view does not render. This is referred to as throttling the query.

The List View Threshold is set at 5,000 because queries that process more than 5,000 rows in SQL Server cause table locks to be applied. Performance for other users will be impacted while this lock is in place.

Note that if the parent site has the *Metadata Navigation and Filtering* feature enabled and you are using the SharePoint browser UI, SharePoint will adjust the query that it sends to SQL Server so that a subset of files is retrieved (typically those most recently modified). This adjustment does not occur when the query is issued via the SharePoint object model, as is the case when using MacroView DMF.

Increasing the List View Threshold will delay the point where SharePoint throttles or adjusts its queries, but the table lock will still be applied by SQL Server and performance will be impacted.

11. Use a fast disk for the Windows system at the client

If Outlook is slow to load in a new Windows session when MacroView DMF is installed or if the first use of *Save As to SharePoint* or *Open from SharePoint* in Word, Excel or PowerPoint takes significantly longer that subsequent usage in the same Windows session, you should check the access speed of the storage used for the Windows system in the client PC.

MacroView DMF is often the first .NET application to load in a new Windows session, which means that the loading of MacroView DMF will trigger the loading of the .NET Common Language Runtime. The CLR is large and will take a long time to load if the Windows system is stored on a slow disk. Typically moving the Windows system to SSD provides an excellent improvement in responsiveness of MacroView DMF (and other .NET applications).

12. Check the overhead imposed by your Virus Scanner

Depending on their configuration, some virus scanners can impose a significant resource overhead at the client PC, which adds seconds to some MacroView DMF processes. An example is Trend Micro Titanium in default configuration, running under Windows 8. When exploring poor client-side performance it is generally a good idea to try disabling any virus scanner.

13. Deploying MacroView DMF via Citrix

Citrix is a very commonly used deployment approach for MacroView DMF, and DMF has been proven to work well via Citrix. However it should be noted that in Citrix terms, MacroView DMF is a Heavy Duty app, because of the large amount of work that MacroView does with the virtual client machine. The Citrix servers should be configured accordingly. 32-bit Citrix servers should NOT be used.

Be careful when using APPDATA folder redirection.

MacroView DMF makes extensive use of the user’s profile. Redirecting the %Appdata% folder to a file share that is not local or on a file server with poor network connection and/or disk I/O will have a significant negative impact on performance.

Feedback from MacroView DMF customers is it is a good idea to utilize Distributed File System when redirecting %Appdata%. This avoids placing a heavy load on a single file share that contains %Appdata% folders for all users.

Setting the MacroView DMF Error Trace to *Information* level reduces performance so ideally should only be enabled when troubleshooting an issue. The reduction in performance is relatively minor is the %Appdata% folder is located on a local disk, but can be severe if %Appdata% is redirected to a shared disk.

14. Use 64-bit clients

MacroView DMF is a .Net application and several of its processes (including file uploads) make use of the Large Object Heap. These .Net processes require contiguous memory so 64-bit operation is preferable because of the much larger memory range that can be addressed.

On a client PC that is running under 64-bit Windows, MacroView DMF Explorer will run in 64-bit mode. However even on 64-bit hardware, 32-bit versions of Microsoft Office applications are commonly deployed. MacroView DMF add-ins will run in 32-bit mode within these 32-bit Office applications. On such machines large file uploads should be performed in MacroView DMF Explorer rather than in Outlook, Word, Excel or PowerPoint.

15. Upgrade to SharePoint 2013

In addition to providing better search, SharePoint 2013 has a number of performance improvements. These include deltas for saving Office files - thereby requiring less data transfer, reducing storage and improving performance of saves.

16. Dedicated WFE for search crawls

Search crawls use a lot of CPU and disk sub-system resources, which will lead to poor performance for end-users if the crawls are running on the same Front Ends as the user sessions. Scheduling crawls to run only after hours is not optimal from a user experience viewpoint – document searches and Search Site Tree commands may not return as many results as the user expects. The better approach is to have a separate server dedicated to crawling and searching.

17. Do not use an under-powered SQL Server

Document management on SharePoint places a heavy load on the Database Server, so you should ensure that the SQL Server is running with adequate resources, ideally on a dedicated server unless the SharePoint document store is small.

You should also ensure that an appropriate SQL Maintenance Plan is in place. A drop in performance after a large volume of new documents are added, or new sites / libraries defined is an indicator that you may need to run Update Statistics. This will update the SQL Query Plan to reflect the new volumes of data. One MacroView DMF customer reported a 50% reduction in response times after they ran Update Statistics!

An effective SQL Maintenance Plan will also truncate log files.

18. Consider using Remote Blob Services

Using RBS is generally more performant than having the BLOBs within the SharePoint Content Database, particularly if using high-speed SAN to store the files.

19. Minimize the need for Inter-site Moves

Unlike the native SharePoint UI, MacroView DMF supports moving documents to a document library located in another site, another site collection or even another web application. The move can be done by drag and drop and as part of the move, metadata is reused and version history retained. However these moves use much more resource than a move within a site. You should not design processes that routinely involve moving documents outside a site.

20. Be Aware of Authentication Overheads

MacroView DMF supports all non-anonymous authentication methods that are available for Microsoft SharePoint. However you need to be aware of the performance overheads that are associated with ADFS authentication. This is particularly the case in wide-area networking configurations, where authenticating a user identity might involve a round trip to an Active Directory server on the other side of the world, adding a performance overhead even if your SharePoint server is local.

More Information

For more information and for consulting assistance with optimizing the performance of your MacroView DMF solution, contact *solutions@macroview.com.au*.

■