Company: **NuView Inc.**
738 Highway 6 South, Suite 850
Houston TX 77079
281 497 0620

ILM Products: **StorageX 5.0**
File Lifecycle Manager 2.0

Submitted By: **Apurva Patel**
Director Marketing & Strategy
apurva@nuview.com

Questions

1. Development Status and Objectives. Describe your vision of D/ILM and identify what features or functions your platform currently provides and what features or functions are still in development.

We have two products that address data lifecycle management: StorageX 5.0 (Sx) and File Lifecycle manager 2.0 (FLM). Both these product focus on Windows-based file data (they do not handle UNIX data or structured data i.e. databases). UNIX development is in progress for both products, and will be completed within 6 months. Sx/FLM can be used as an integrated solution.

Sx is data management software that logically aggregates file data spread across distributed file storage devices using a Global Namespace, and then allows IT to implement a wide variety of policies that together achieve file lifecycle management in the broadest possible context. While Sx acts on files, policies get implemented at a folder level. These policies include:

- Capacity utilization efficiency through data migration based on age of files, size of files, access frequency (without having to leave behind stub files or pointers, as the Global Namespace updates file locations on-the-fly. This single feature makes Sx a completely transparent lifecycle management solution)
- Capacity allocation efficiency through automated load balancing of data among storage devices in a pool of storage – a pool can be setup by policy that lets you drag and drop devices in the network to populate the pool
- Business continuance through replication of files based on policies that let you prioritize data meant to be replicated, followed by instant failover of users to alternate storage devices/locations in the event of a disaster
- In-depth data classification using physical and logical categorization to enable IT to make decisions for implementing lifecycle management policies

FLM is a more focused solution that seeks to optimize a large Network Appliance file storage repository. Note FLM's architecture enables IT to use all its selection parameters in combination ('and/or' format as compared with 'or'). Policies in FLM get implemented at the file level, and include:

- File classification based on file age, size, type, location, attributes and volume utilization
- Simulating different data management actions based on a mix of the above parameters
- Migrating data based on above parameters to one or more storage devices and retaining a stub file on the source storage device (note this file contains all the metadata of the original file and is not a 'shortcut')
- Blocking certain file types from being created on the primary storage devices
- Deleting files that meet a combination of the parameters listed above.

2. Development Partners. List the vendors with whom you are working to deliver your D/ILM solution functionality.

Sx works with file storage devices from Network Appliance, Dell, HP, EMC, and other Windows Storage Server based appliances. FLM works with Network Appliance platforms as the source of data, and any Windows-accessible storage as the destination.

3. Data Naming Scheme. What is your method for identifying data storage requirements or characteristics and for using those requirements or characteristics in building policies?

Sx allows administrators to create customized 'logical' groupings of their data. It does not directly assist in identifying data storage requirements, but does let the administrators perform lifecycle management using any number and type of logical groupings (department, user, project, location, tier of storage, etc). The key thing to note here is Sx allows IT to create multiple groupings or views for a single dataset (one would normally have to make multiple copies of data to present it in multiple ways).

FLM allows IT to perform simulations based on multiple selection parameters, and generates reports based on the results of the simulations. IT can use these reports to make decisions like 'what data should be allocated to what storage'. FLM allows any number of policies to be setup, and enables the data output of each policy to be migrated to different storage devices.

4. Access Frequency. Identify the mechanism by which your solution identifies how frequently specific files or datasets are being accessed and whether and how access frequency is used in migration policies.

Policies in both Sx and FLM allow IT to classify data that has not been accessed based on the 'last access date' Windows timestamp.

5. Storage Platform Characterization. Explain the mechanism that your solution provides for characterizing the performance capabilities and costs of specific hardware platforms for data storage so that this information can be used to target the appropriate storage platforms as destinations for automatically-migrated data.

Neither Sx nor FLM characterize the performance of hardware, but allows administrators to setup pools of physical storage as well as pools of data once they have chosen the storage and data, respectively.

6. End of Useful Life. Explain how your platform facilitates the automated removal and clean-up of data that has outlived its useful life and restoration of freed capacity for use by applications.

FLM has a deletion policy that analyses data based on parameters described in point 1, and deletes it based on a pre-set schedule or on demand. In addition, FLM provides a safety mechanism whereby administrators can make a copy of the file on some second-tier storage prior to deleting it from primary storage.

7. Policy Articulation. Describe how policies are created and how they are applied to existing data.

Policies in Sx can be created at the folder level, and there is no limit to the number of policies for each type of data management action (replication, archival migration, load balancing, failover, etc.). Policy creation is highly intuitive and accommodates drag and drop mechanisms. Policy setup and execution in FLM is as easy as in StorageX, except that policies in FLM get setup and executed at the individual file level, while policies in Sx get created at the folder level.

8. Device Support. What storage devices does your product support? Are their any proprietary devices (controllers, arrays, HBAs, switches, SAN topology, virtualization products, etc.) that are required for your solution to work?

Sx uses a standard Microsoft technology called DFS to effect the virtualization, while FLM leverages a specific interface made available by Network Appliance. Both products work with CIFS (industry standard) accessible data.

9. Resource Consumption. How much bandwidth and server CPU "overhead" is introduced by your solution (e.g., to support polling processes, migration processes, agent processing, access frequency counting, etc.).

Sx is an out-of-band solution and therefore causes no CPU activity for either clients or servers. FLM runs on a dedicated Windows server and is architected to minimize interaction with the source device (NetApp filer).

10. Flexibility. Does your solution have multi-vendor support? Can data, once integrated into your D/ILM scheme be migrated readily between your solution and other solutions in this space? Is you solution interoperable with other solutions in this space?

Sx works with file storage devices from Network Appliance, Dell, HP, EMC, and other Windows Storage Server based appliances. FLM works with Network Appliance platforms as the source of data, and any Windows-accessible storage as the destination. Our solution is interoperable with other Windows (CIFS) based solutions.

11. Speeds and Feeds. How should a prospective customer compare the relative performance of competitive solutions in this space? What are the appropriate performance measures or metrics to use in evaluating competitive solutions?

We feel customers need separate solutions for three key data sets (files, databases, special applications). Within each of these categories, the following aspects need to be considered closely:

- Transparency to users
- Breadth of coverage in terms of storage platforms
- Ability for policies to be standardized across the organization
- Standardizing the unit of evaluating data independent of the underlying storage platform
- Number and depth of classification parameters
- Types of actions that can be performed in addition to plain migration
- Ease of use and implementation
- Location independence

12. Interconnect Support. List the network or fabric interconnects that you support. Is there an optimal interconnect for data movement for your solution? What about for management?

Standard Ethernet.

13. Protocol Support. List the storage networking technologies supported, including FCP, Ethernet, iSCSI, FCIP, iFCP, Parallel SCSI, SAS, and network file system protocols (NFS, CIFS, HTTP, DAFS).

Ethernet is the transport, and CIFS is the key protocol. NFS support will be added this summer.

14. Data Type Support. List the data types supported by your solution, including file types and file systems, databases, and hybrids such as email. Comment on the granularity of your D/ILM solution: will it support the migration of database components or subsets, subsets of email files, etc.?

Windows files (not databases). Both Sx and FLM will act on individual files.

15. Cost. List the cost of your solution or provide some means for calculating cost for a specific environment.

StorageX lists for $2000 per node, and FLM is priced by type of NetApp filer.

16. Standards. List any relevant open standards upon which your product is built.

Sx is built on top of Microsoft DFS (distributed file system)

18. Miscellany: Please note any additional information that you think would be worthwhile for prospective customers to consider about your solution or other solutions in this space.

Today, both storage equipment and management processes lack standards. This is the principal reason why meaningful TCO reduction will remain distant. StorageX uses a Microsoft technology called DFS to built its virtualization infrastructure. DFS is present in every Windows client and server sold for some years now. Therefore, StorageX presents a non-intrusive standards-based method to implement lifecycle management for files. Other vendors of data lifecycle management software must install agents on or close to storage devices to implement their policies, as they have no unifying underlying platform such as the Global Namespace. It is very important for clients to consider the positive implications of standardizing data management processes, which is only possible when data management activities are implemented off a standards-based platform, as compared with implementing piecemeal agent-based solutions that do not share meta-data and other intelligence.