

# EVALUATING z/OS SRM PRODUCTS

“Realities,” Part 2

## UDM WEBINAR

August 24, 2017 1:00pm EST  
Get a live demo and learn  
how simple z/OS storage  
management can be with  
UDM.



**Dino-Software**

## Realities of z/OS SRM Tools - Part 2

Part 1 explored primary implementation aspects to be considered when evaluating SRM products to manage data on z/OS Systems. The complexities of the installation and maintenance process were dissected, and points were highlighted to consider during that phase. Part 2 continues the assessment into top ways in which an SRM Tool should help an organization solve problems.

### How an SRM Tool helps Solve Problems

An SRM Tool can become a key asset to an organization if it provides a solid return on investment by helping streamline, optimize, and protect operations. An SRM tool should provide significant contributions in managing enterprise systems and hardware independent support for storage objects. Many tools are so complicated and disparate that they are difficult to learn. This can lead to the continuation of inconsistent processes and procedures, lack of standardization, and sometimes a waste of money when the tool stays on the shelf and does not get used.

These aspects should be considered in the evaluation and selection process:

### *Simplicity – simplify selection, view, and management of data*

One of the major drawbacks with most SRM tools is not just the complexity of installation but also the lack of intuitiveness in the product.

If a user simply wants to run a report and get a list all Storage groups on all Systems that are over 80% used, it's not a simple process but very convoluted. This point alone leads many SRM tools to a life of occasional usage, rather than them being at the forefront of z/OS management. An SRM tool needs to be simple to use, and the following questions and points should be contemplated:

- Can the data be summarized in any way the user would like it? For example, using a dataset list as input, can the user easily show how much data is associated to each unique creation date? Or are they limited to boxed summary reports provided with the product?
- Can the data be filtered in any way the user would like? For example, can a user find the needle in the haystack, or are they limited to a basic filtering process with limited multi clause SQL type filtering options?
- Can the data be easily cross referenced with other objects? For example, when in a Storage group report, can the user then:
  - See the volumes in the Storage group
  - See the datasets on the volumes
  - See the associated meta data for the datasets, such as Catalog entries, VVDS entries, and DSCB fields.
- Can the historic status of an object easily be seen? For example, when showing Storage group usage, can the user easily see the usage of the Storage group over a previous time frame, and can they, for each object, choose how much history to retain?
- Can the data easily be used as input to z/OS or TSO commands or JCL decks to allow the repair of a problem? And if actions are initiated in this way, can their progress be followed without having to return to TSO?
- Can users really do all the day-to-day work from the SRM tool, or will they end up switching back to TSO to perform certain tasks like initializing volumes?

## DINO WHITEPAPER: Realities of z/OS SRM Tools - Part 2

### Usability

Another important aspect is the overall usability. Was the product designed to merely achieve functional goals, or was it designed from the ground up with the perspective of a storage administrator?

There are several factors to consider in determining if a product can help users effectively and efficiently do their job:

### Standardization

Any product is going to have a learning curve, but if that learning curve occurs every time a new version is released, it becomes problematic.

If a proprietary filtering or scripting language is used and it changes, not only is there something new to learn, but the existing saved queries must be modified. Using a standard filtering process such as SQL reduces this impact. Additionally, any tool needs to provide a homogeneous process for querying the status of objects; for example, querying catalogs should be the same as querying another object such as tape usage, with the only difference being the data being analyzed.

### Portability

One of the main goals with an SRM tool is centralization. Instead of having copies of reports, JCL decks, commands or automation processes spread all over the place, there is a central location where they are defined. When that process needs to be run, the request can be routed to the required systems.

If, due to differing host/GUI versions, the user is forced into having multiple SRM Enterprises, they will have to duplicate these processes by saving the definitions in each enterprise which actually reverts to keeping multiple copies.

The second issue is when defining automation, such as message traps or timed schedules, to go off and create reports, what happens when a new system or customer comes online? How do these processes get ported over to new Systems, or how can customizations be shared with others or with a User group?

Portability should mean anything being done can be ported to anyone and anywhere via a simple offload/reload type option.

### Flexibility

One size does not fit all when it comes to SRM, and the product needs to allow the user to decide the format, etc., of reports.

A classic example is space formats: some users might prefer values in Mbytes, others in Mebibytes, and others in Cylinders. If the product forces a specific format, users don't have flexibility to specify their preference in which the metric space values will be displayed.

### Speed

Web interfaces especially tend to show limitations when a large amount of data is being analyzed. This is because either the bandwidth used to get the data to the GUI is limited, or the design of the interface means only a subset of the data is visible at a time and never the whole amount; for example, a single page of data is displayed, and the user clicking page down initiates collecting the next page of data. Sometimes the frustration of not being able to quickly and easily scroll through the results leads a user back to good old ISPF 3.4. This limitation becomes more apparent when they want to offload the data; for example, an Excel spreadsheet, as, at that point, the whole amount of data needs to be uploaded to the GUI, and then the real limitations of the product are realized.

Even if the speed is there, is it really what the user thinks? Is the product actually analyzing the current environment, or is it analyzing a snapshot taken at some previous time?

## DINO WHITEPAPER: Realities of z/OS SRM Tools - Part 2

Many products have collection processes which scan the Environment in the background and then allow the GUI to analyze those point-in-time scans. Sometimes this is appropriate, but, for example, if doing cleanup work on catalog entries, dynamic information is needed to reflect the status of the environment after the cleanup is performed.

More importantly, a user needs control over this, and they need to know if the data is from a previous time zone. In many cases, this small but important piece of information is not as apparent to the user as it should be.

### Enterprise Wide

A huge limitation in most SRM offerings is the inability to obtain information from multiple Systems or sysplexes in a single request and single view. Some provide limited support, but if a true Enterprise tool is the requirement, it needs to be standard. Whatever action is being taken, the option should be there for the user to select any system(s) where the request should run, and the results should be returned to a single window.

### Completeness

Whenever a GUI is used, a common question is, "can I do everything from the GUI, or do I end up having to go back to the green screen?"

This will occur because of two reasons:

- The user is not sure if something actually worked because the GUI is not giving them the information needed, and they confirm this by switching over to TSO.
- The GUI provides an incomplete solution. For example, if a JCL deck is fired off from the GUI, it does not allow viewing the status of the job or the output from the Job.

A truly complete SRM product will allow the user to initiate, monitor, and review a job entirely from within the GUI.

### Conclusion

Archaic solutions, which are being retrofitted into today's environments, coupled with overly complicated installation procedures have resulted in SRM products which are having little real term and practical use and often stay in the box.

Before signing up for an SRM tool or extending the license on an existing tool, perhaps now is the time to ask some serious questions:

- Is it an Enterprise solution, performing tasks on multiple z/OS systems simultaneously, or in reality, will users end up still working at a system-by-system level?
- Is it easy to install and maintain, or due to the complexity of the product, is it going to result in complex network customizations to ensure the GUI has access to all the systems it requires and excessive RACF customizations to ensure the SRM servers have the required access to get the job done?
- Is it a common interface, or in reality, is it actually a group of heterogeneous products with a common name, meaning an increased learning curve and limited interaction between the products?
- Is it secure, or does it bypass the security infrastructure, as access to the SRM engine ensures access to anything rather than only to things a user normally would have access to?
- Is it strategic, for example, filtering using SQL, or is it using some proprietary process which requires more learning and is prone to change between versions?
- Is it worth the investment, or will the business spend more time and money installing and maintaining the product than it would using the old-fashioned methods?

### Dino-Software's Solution

The holy grail of z/OS storage management is a true Enterprise wide solution for managing all storage on all

## **DINO WHITEPAPER: Realities of z/OS SRM Tools - Part 2**

Systems and Sysplexes via a common, easy to use and install, heterogeneous interface. All of this is exclusively available within UDM.

Dino Software's Universal Data Manager (UDM) is a recent entrant into the z/OS SRM market space, and it encompasses all the essential aforementioned components. UDM takes advantage of today's development architectures and makes management of the Enterprise as simple as managing an individual system.

Using industry standards such as SQL filtering and REXX based automation and ensuring integration into an existing security infrastructure, the product ensures quick and effective management of an Enterprise from a single GUI window.

When it comes to SRM, UDM keeps it simple. Take a closer look at UDM by visiting [Universal Data Manager \(UDM\)](#).