## Universal Data Manager (UDM): Enterprise Level z/OS Storage Management GUI

Universal Data Manager (UDM) is a GUI based solution for managing your z/OS Storage at an Enterprise level including; ICF Catalogs, DASD Volumes, DFSMS objects, Spool data and other related objects. The overall ease of installation and automated analysis provides the controls, information and reporting needed to manage the dynamics of modern System z Storage environments.



Using the UDM GUI interface you can:

➢ **Automate** the management of z/OS objects by dynamically reacting to Console messages or conditions.  Additionally, you can schedule timed processes to interrogate the status of an object.

➢ **Monitor** the status of z/OS objects via the centralized monitor, which allows you to view and react to alerts from all z/OS Systems via a single GUI window.

➢ **Report** on all aspects of your z/OS Storage subsystems centrally, in real time or historically, using SQL filters. For example, using a single request you can view in a central window the status of Storage groups on all your z/OS Systems.

➢ **Maintain** your z/OS Systems using the Command or JCL interfaces which allow you to centrally define store and run Z/OS commands or JCL decks.

Simplified installation, minimal customization and pre-configured processes mean you can install and gain the benefits from the product in minutes rather than days.

## Why Is UDM Necessary?

Management of z/OS storage is becoming more of a challenge due to the ever-increasing size of Data Storage environments and the decrease in the manpower to Terabyte ratio to manage this Storage. The net result is critical management processes such as Audits of the Catalog environment are being ignored whilst other tasks such as the initializing of volumes are performed without the time to ensure the required checks have been performed, thereby increasing the possibility of user errors.

UDM solves these issues by simplifying and automating most of the tasks involved in managing your data.

Distributed with UDM are a series of processes, which collectively provide an out of the box solution that will:

➢ *Collect* information on objects and retain Historical information allowing proactive analysis of data to determine any actions required.
➢ *Simplify* the processes needed to report on an object via a series of boxed reports.
➢ *Trap* critical Console messages and alert you when critical conditions such as DASD errors occur.
➢ *React* to object conditions such as Catalog or VTOC errors.
➢ *Notify* you of exception conditions as they occur via the central Monitor window that receives alerts from all your Systems.
➢ *Simplify* the day to day processes such as initializing volumes by ensuring volumes are offline to all systems, including systems outside the sysplex.

Additionally, UDM provides the ability to define and test your Service Level Agreements on a single system and then via a single mouse click, provides the ability to Propagate the processes to manage these service levels to all your Systems.

## Who Is UDM Designed For?

UDM fits seamlessly into any Data Center due to its flexible architecture. Customers with existing third party products can easily tailor UDM to interface with these existing products.

Irrespective of the Data Center size UDM provides immediate benefits:

➢ *Large Multi Customer* sites can use UDM to define a standard set of Service Level Agreements and then propagate these to existing customer Systems, via a single mouse click, or implement these standard Service Level Agreements as new customer Systems come on board.
➢ *Big Data Centers* benefit from the ability to work with multiple systems at the same time, for example defining a new ICF Alias for a new User can be performed on all Sysplexes in a single request.
➢ *Small Data Centers* with less manpower resources benefit from the pre delivered and configured processes meaning fewer resources are needed to maintain and monitor Storage, thereby freeing up personnel for other tasks.

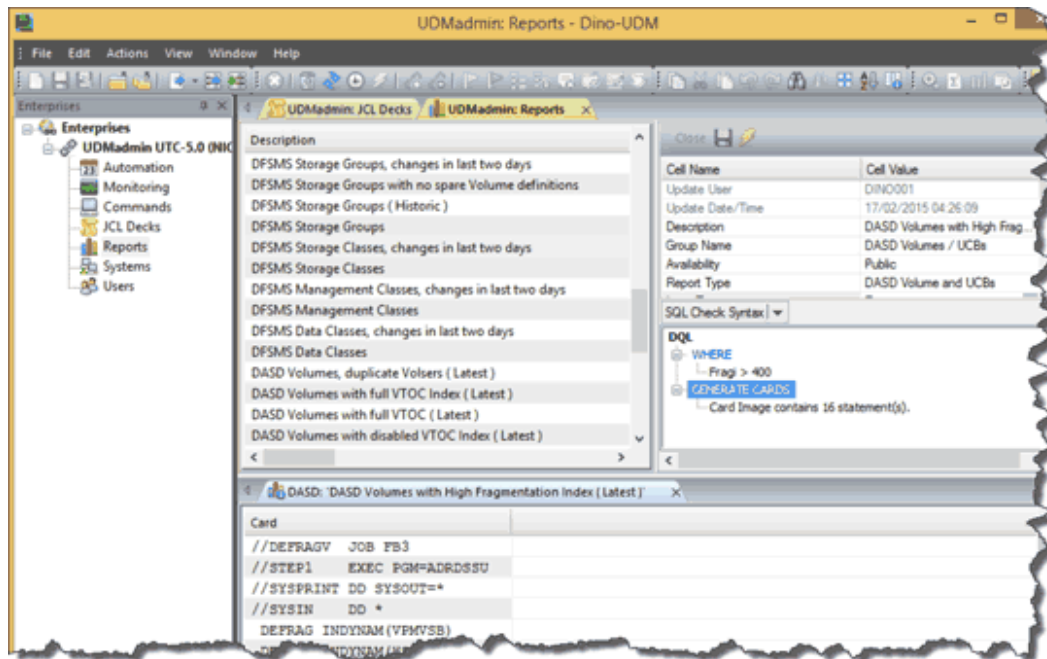Storages Managers, Capacity Planners, Operations Support and Systems Programmers can all benefit from UDM features:

➢ *Storage Managers* can use the powerful monitoring facilities to be alerted when critical Storage conditions occur such as full Catalogs or Storage groups. They can then utilize delivered Commands or JCL to react to these conditions or customize the product with their own, site specific, actions.
➢ *Capacity Planners* using the SQL based reporting processes can automate the collection of capacity related data such as Usage by High Level Qualifier and easily review and manipulate this data.
➢ *Operations Support* can easily analyze problems such as volumes accidently being offline to systems and react using customized Commands or JCL decks.
➢ *Systems Programmers* can easily analyze System messages, Spool or the general Storage architecture and issue z/OS maintenance commands via the Commands Interface.
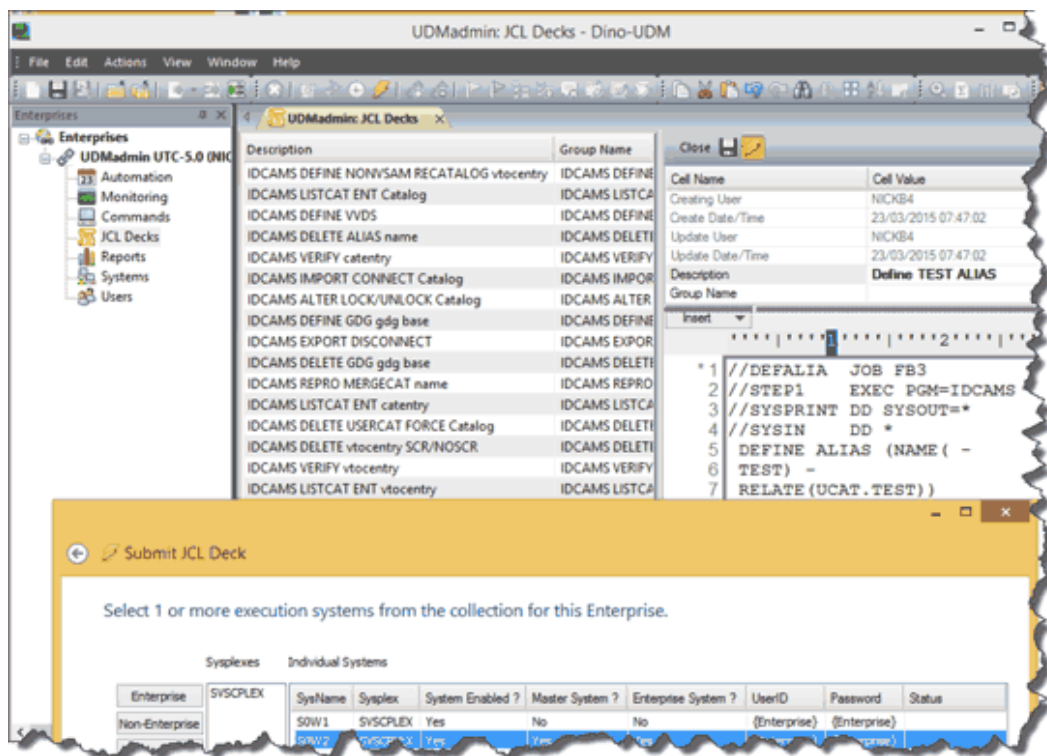
## When Should UDM Be Used?

UDM can be used to proactively monitor your Systems or it can be used to perform ad-hoc tasks on request. By activating UDM on a System, simply by starting the Started task, UDM will be automatically monitoring and maintaining your Systems as well as collecting historical information on your System for further analysis. Once active UDM provides a comprehensive set of processes, which allow you to perform common tasks such as the following:



*Identify DASD volumes with a high Fragmentation Index, then build & submit a DFDSS DEFRAG job*



*Generate & submit an IDCAMS DEFINE ALIAS job on all Sysplexes to define a new Userid*

## When Should UDM Be Used?



*Identify high capacity users of JES spool, grouping by Ownerid and Summarizing by Record count*



*Automatically issue a DFHSM SWAPLOG command on all systems at midnight*

## How Can UDM Reduce TCO & Save Money?

➢ *Simplify and expedite the implementation of new storage policies.* Allow for changes to be propagated to your Storage management SLAs and processes to new z/OS systems via a single mouse click.

➢ *Centralize and simplify Storage Management processes.* Allow for changes to be propagated to all systems in your Enterprise via a single mouse click ensuring a dramatic reduction in resources needed in managing multiple Systems and Sysplexes.

➢ *Automated data collection for better decision making.* For example, when more DASD storage is required or better still, identifying wastage in your environment, which if cleaned up would negate the need for further storage acquisitions.

➢ *Eradicate risks and consequences of errors and omissions.* For example, space failures, ICF catalog full scenarios, failed backups, et al. All scenarios that are otherwise avoidable with advance notification and resolution with UDM.

➢ *Reduce and simplify your Software inventory.* If you have individual products to report on parts of your Storage subsystem or products to automatically defragment volumes or products to identify and repair errors UDM would allow you to reduce your Software costs as it can perform all these functions and more in a single product.

➢ *Reduce the manpower needed to manage your z/OS Systems.* If you have multiple Systems and Sysplexes and want to clean up the datasets for a TSO User who is leaving the company it is likely you will need to log onto each Sysplex and manually list and remove the Users data. Using UDM a single action could list all the Users data from all your Systems and Sysplexes and a subsequent Delete action could remove all the data in a single mouse click.

## How Can We Deploy & Use UDM?

The major design objective of UDM is to allow simple resource efficient deployment and usage of the solution, allowing the user to quickly deploy and benefit from comprehensive product function:

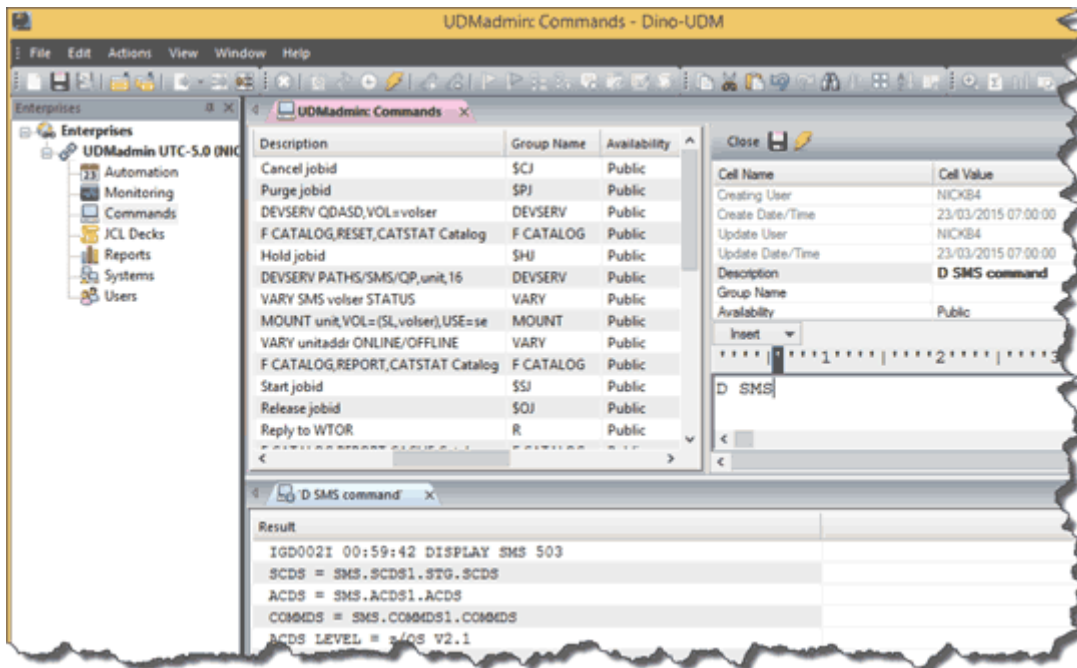## Simplified Streamlined Installation & Integration Process

➢ *z/OS Host Installation* requires three libraries, including one PARMLIB per Sysplex, configured with five parameters and the UDM Started Task (STC) to be started.

➢ *Client GUI Installation* requires the GUI.EXE file to be loaded into a folder shared by all potential UDM product users.

➢ *Seamlessly Integrates* with existing security products (E.g. ACF2, RACF, Top Secret), where all actions are processed using authorization from the calling user, not the authority of the UDM Started Task (STC).

## Simplified z/OS Command Interface

➢ *Allows z/OS Commands* to be centrally stored in a single location and then issued to multiple Systems in a single request.

➢ *Ensures Command Output* from all Sysplexes is returned to a single GUI Window.

➢ *Allows Generic Command Decks* to be created via the use of Variables. For example a Vary command could have a variable Unit Address that would result in the user being requested to specify the Unit Address prior to generating the command.

➢ *Safeguards Console Commands* are submitted using the authority of the caller and not the UDM Server.

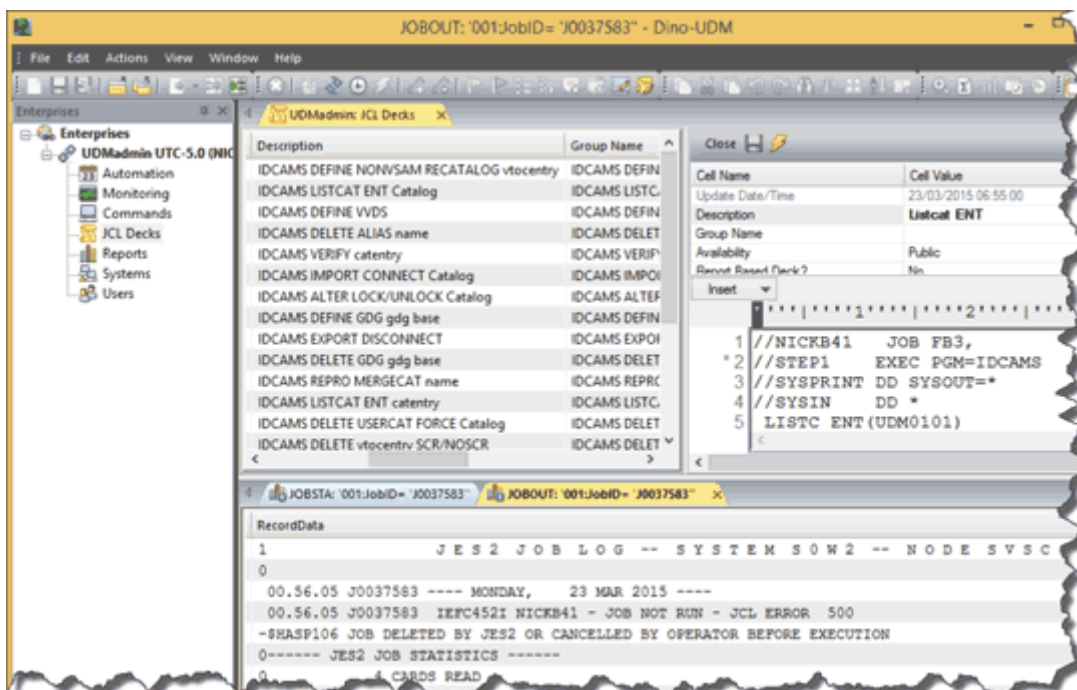## Simplified JCL Management

➢ **Centrally Stored JCL** decks in a single location for submission to multiple Systems in a single request.

➢ **Integrated Spool Management** process allows you to review the status of Jobs, including output from multiple systems in a single GUI window.

➢ **Generic JCL** deck creation via the use of Variables, for example an IEBCOPY compress dataset JCL deck could have a variable Dataset Name, which would result in the user being requested to specify a Dataset name to compress prior to generating the command.
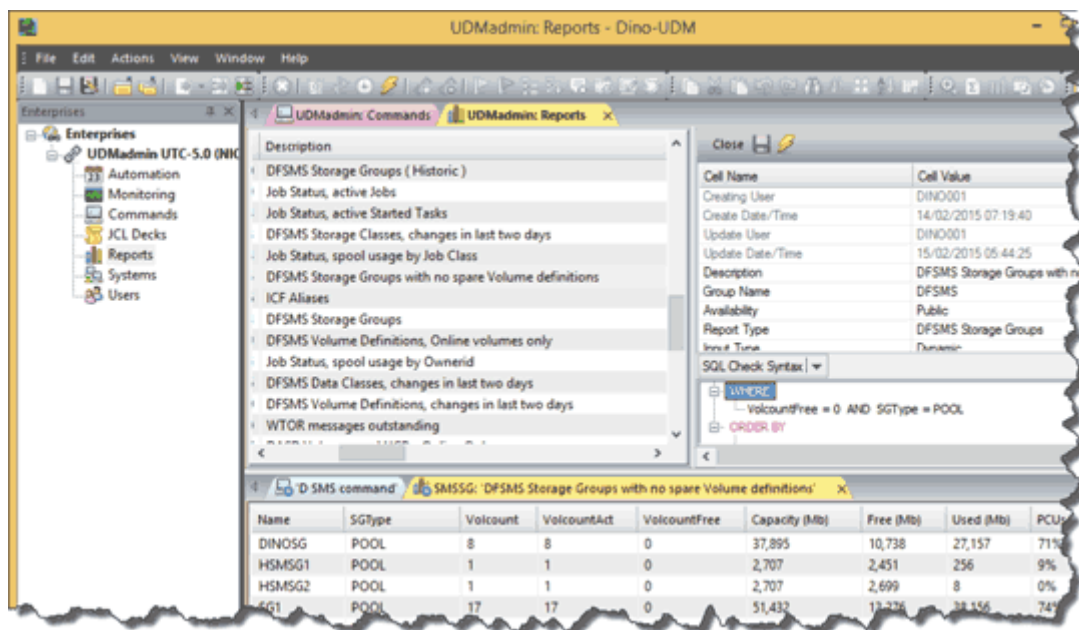
## Simplified Reporting Interface

➢ **Report** on a specified Object across multiple systems in a single request. Results from all Systems are returned within a single GUI Window.

➢ **Full SQL Filtering** ensures Report data can be sorted, filtered or summarized using standard SQL syntax meaning complex filters can be set up to identify specific conditions or Group and Sum features can be used to generate Summary reports.

➢ **Drill Down Functions** ensure easy cross reference between reports for example when listing Storage groups you can easily list the Datasets on the selected Storage group(s) or Volumes within the selected Storage group(s).History Reporting functions simplify predictive analysis, For example, when displaying Storage groups, selecting a Storage group with the History option, displaying historic information about the selected Storage group.

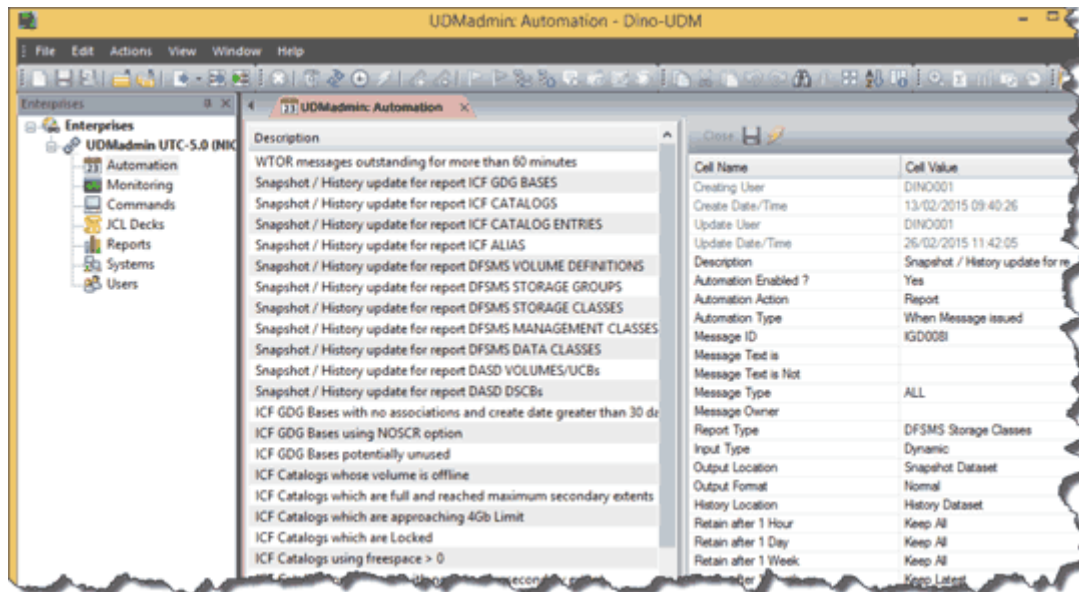➢ **Action** functions allow Report data displays to be used as input for JCL or Command skeletons.



## Enhanced Console Automation Interface

➢ **Trap** console messages based on ID, type or text within the message.

➢ **Process** automation, as per conditional message processing; including Operator Command processing, JCL deck submission, Report processing and/or Message notification via the central Monitoring interface.
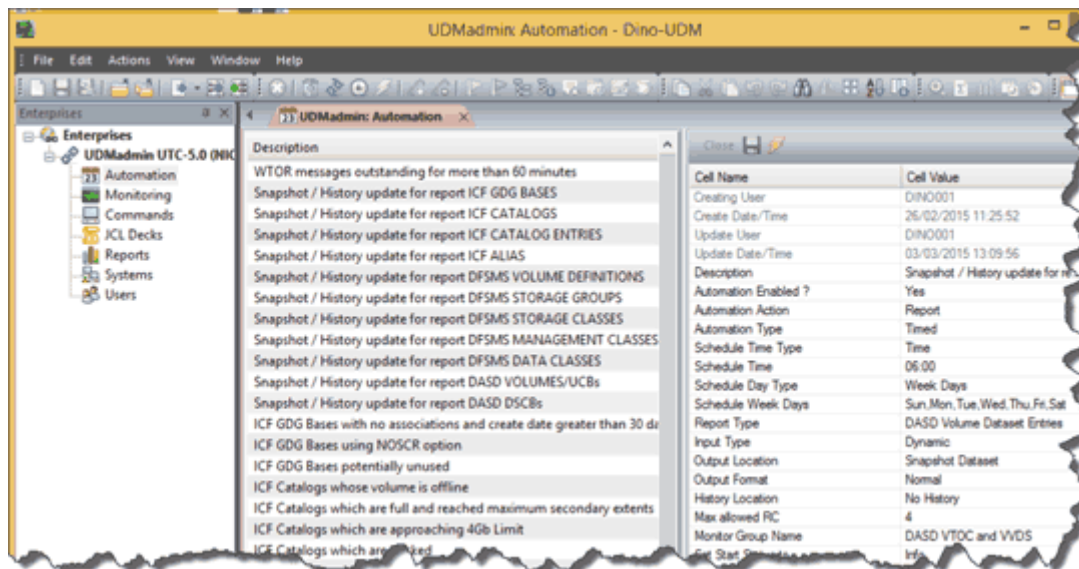
## Enhanced Process Scheduling Automation Interface

➢ **Schedule** processes at specified times, when existing UDM controlled schedules or independent jobs end.

➢ **Processes** include Operator Command processing, JCL deck submission and report processing. Additionally dynamic repair actions can be generated and submitted. For example, a timed process to identify DASD volumes with a High Fragmentation index which would then generate and submit an appropriate DEFRAG JCL deck.
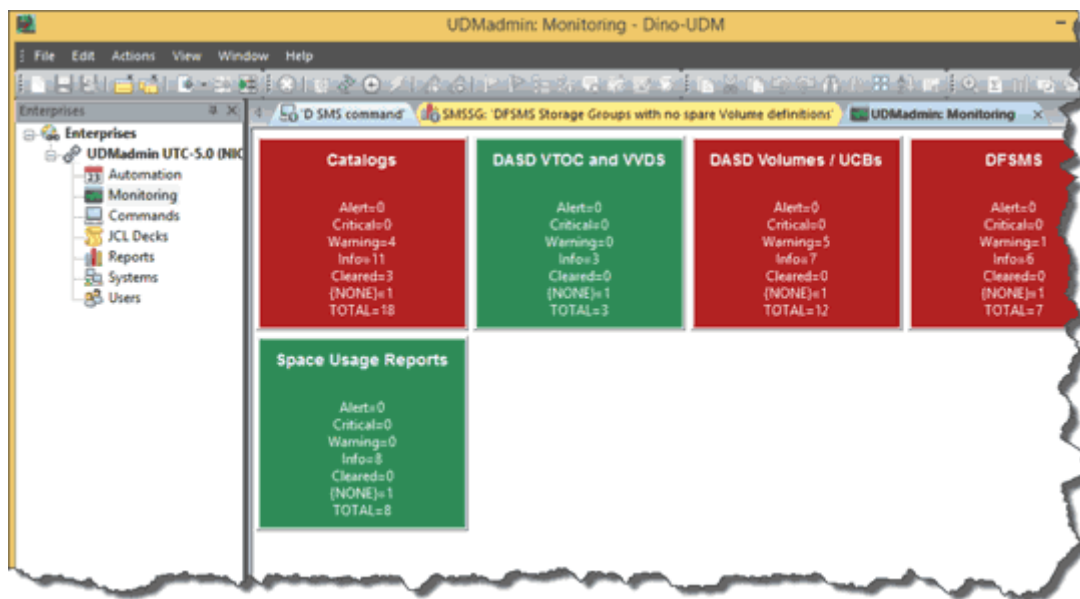
## Enhanced Exception Condition Monitoring Interface

➤ *Receives* UDM driven alerts from all systems in single GUI window.
➤ *Defined Alerts* can be based on a message being issued, a job or process ending or as a result of a Report query returning data. For example a report process for locating Catalogs, which are full.
➤ *Targeted Alerts* can be defined for specific scenarios. For example Catalog alerts.
➤ *Drill Down* capability allows you to identify issues, for examples catalogs with issues. Then view a problem summary, for example all full Catalogs and then identify the actual Catalogs, which are full.



*For more information please visit the UDM Portal or Email our Sales Team or call us on +44 (0) 845 0579386.*