

KAMPGROUNDS OF AMERICA



As North America's camping leader, Kampgrounds of America (KOA) offers campers the chance to escape their daily routines and enjoy their freedom in unique, scenic places at any of North America's 450 KOA Kampgrounds.



Data Recovery and Availability at a higher level

Like many other iSeries users, KOA has seen a dramatic change in the way that computer systems are used by the business. Instead of being used only by 'internal' users, our system is now supporting a greater variety and quantity of both users and applications. This growth has occurred as part of our strategic goal of using technology to leverage our competitive advantages in our markets.

As our hardware environment grew to keep pace with growth demands, our staffing level remained about the same. We have a small shop. Five people have iSeries application development experience on the iSeries, along with a Network Administrator and a Systems Operator. Installing the new systems, developing, testing, and documenting the new applications, and assisting over 200 franchisees in connecting with the new server(s) consumed all available staff hours, leaving us very little time to devote to systems administration tasks. During this period we had continued with the 'basic' data recovery that had been in place since the original AS/400 was installed .saving everything to tape each night, and taking advantage of device parity protection to avoid losing data if one drive failed.

With the growth in the volume of transactions entered each day it was obvious that we needed to move beyond relying on a 24-hour-old tape backup in terms of recovery. Unfortunately, we could not afford either the time to program a solution ourselves, or any of the packages available from the major High Availability vendors. Then, we discovered the *noMAX application package, written and supported by a New Zealand company, Maximum Availability Limited.

*noMAX was priced considerably lower than other vendors

This package was priced substantially lower than the quotes we had been getting from other vendors, and appeared to be in our price range, so we decided to go ahead with a project to back up the data from our iSeries model 810 server to the iSeries model 270 server. Our goal at this time is not to have a 'hot fail over' capability, but to provide an up to the minute copy of the database, which can be used to run reports and to reduce the backup downtime on our primary server.

All of the High Availability solutions are based on Journaling, which records changes to files and other objects as they happen. One of our first challenges was to learn about and implement Journaling on our production system. In the past, Journaling has had a bad reputation as a disk performance hog.

We had already run into disk performance problems once on a prior AS/400 server, which had I/O controllers with a 4MB write cache. This painful experience made us very concerned about the impact of Journaling on our system. IBM has excellent Redbooks dealing with Journal performance, and the staff at Maximum Availability also helped us overcome our fears.

Following the recommendations in the Redbooks, we ordered additional drives to set up additional parity sets and user ASPs on both the production and backup servers. We had also configured our model 810 with a disk controller with additional write cache (over 100MB instead of 4MB). The disk controller of the model 270 has 26MB of write cache. By buying the additional drives on the used market, we kept our hardware expenses minimal.

*noMAX fully utilises Journaling and Remote Journaling

Once we had the hardware in place we then had to set up Journaling on the files we wished to have backed up. In the process of determining what to journal we discovered just how many 'temporary' files existed on the system. These files had to be investigated to see if they a) should be journaled, b) were still being used by an active application but didn't require Journaling, or c) could be deleted. This part of the project took longer than any other step. Once we had determined what to journal, we had to create the Journals and Journal Receivers; and then a CL program to add the files to the Journal we just created. This part took only a few minutes of programming.

After setting up Local Journaling, then next step was to implement Remote Journaling. *noMAX uses the Remote Journal support built into the OS/400 to move the Journal transactions from the source system to the target system. In theory, this is a simple exercise to set up; using a TCP/IP connection over our network. In our implementation we defined a new interface on each server, in the non-routable address range (192.168.), and then enabled a remote relational database on the target system. Once the systems were communicating and aware of each other, we could create the Remote Journals. This process took only minutes, and was expedited by the detailed instructions provided by the *noMAX Installation Guide.



Case Study

In addition, we now had to synchronize the operations of the source and target servers. For Remote Journaling to work, both servers must be up and running including a TCP/IP connection between the servers. Our backups and IPL schedules on the target server had to be modified to meet the system availability constraints of the source system. In our daily backup process we shut down TCP/IP connections while the save was running, to prevent object locks from interfering with the save.

Our target server IPL'd on Saturdays, while the source server IPL'd on Tuesdays (Saturday being a busy time on the campgrounds which used the source server!). We also added steps to the backup process to shut down the Remote Journaling, change the Journal Receivers; and then restart TCP/IP and the Remote Journaling when the save was finished. Once the schedules were synchronized Remote Journaling could be counted on to copy all transactions from the source system to the target system in a reliable manner.

The evaluation application became our production application

Now that the Journal entries could be replicated to the target system, we were ready to install and configure the *noMAX application to 'play' the Journal records against the backup copy of the database. We downloaded the evaluation version of the software, and installed it on both the source and target systems. The evaluation package is the production application, with a 30-day trial license key. All of our setup was done during the trial period, and it became a production application when we received the permanent license key; with no additional work required. Once the software was installed we were given a quick tour of the features and how to configure the application. This was done by the technical support staff at Maximum Availability using WebEx to show us the online 'live' product demonstration. This allowed our support staff to see the screens on our PC, which was connected to the iSeries using terminal emulation. They could also see the GUI interface screens used to configure the *noMAX application.

*noMAX comes with native GUI or standard OS/400 command interface

After the software demonstration it was time to create a configuration using our files and see how it ran. The *noMAX application has both a fully native GUI interface, and standard OS/400 command interface for configuration and monitoring. We used the command interface to configure the files to be replicated, so we could automate the process (we had over 3500 files to replicate). The programming for this was again very simple, using examples from the *noMAX Operations Guide provided with the software.

We have achieved our goal of having a backup copy on a separate server with minimal impact on the overall system

With the configuration in place it was time to start replicating the data. This requires a 'checkpoint save' on the source system (while the files are being journaled). This copy of the data is then restored to the target system; and the *noMAX application reads through the Remote Journal entries until it finds the 'save' Journal entry. At this point it begins applying the Journal entries in the Remote Journal to the backup copy of the database, producing a mirrored copy of the data files. (Note that the *noMAX product also mirrors other objects the same way using custom Journal entries, providing for a full High Availability solution, but we chose not to implement these features yet). The 'checkpoint save' was the only function which had to be done with no other applications accessing the database, all the rest of the setup could be done without

With the data restored to the target system, and the Journal entries being applied by the *noMAX software we have achieved our goal of having a backup copy of the database on a separate server. We are now in the process of evaluating the new capabilities that the software provides us, including eliminating the backup process on the source server; as well as being able to run some of the nightly reporting jobs against the backup copy of the database; relieving some of the workload on the source system.

Happily, all of the Journaling, Remote Journaling, and *noMAX software applying changes to the backup database has been accomplished with minimal performance impacts. Our fears about Journaling support were based on outdated information, with the new DASD controllers having a minimum of 26MB of cache eliminating all of the problems we had expected. The Remote Journaling and *noMAX software run completely on the target server, and do not impact the production (source) server at all. Even on the target system the resource use is very low, and has had a minimal impact on the overall system.

All the above has been accomplished at a minimal cost in programming resources (not counting the system cleanup time), and was supported through the entire process by the excellent technical staff at Maximum Availability.



"I highly recommend to anyone who is not yet journaling or does not have an installed High Availability solution to get started now. It is easier to do and less expensive than you think - and the cost of NOT doing it is too high for you or your company to afford!"

Jef Sutherland

Vice President, Information Services
Kampgrounds of America

