

## **Distributed Computing Infrastructures and Validation**

By  
Pamela Campbell  
DataCeutics, Inc.

The distributed computing paradigm is infiltrating every aspect of the pharmaceutical industry. In the past, an application was confined to a single server or mainframe. Within today's new client/server distributed computing architecture, applications no longer run on a single system but are spread across a network. These new distributed applications span not only servers but also operating systems and disk farms. This new computing model has created new validation challenges, necessitating the need for system management and operation teams to become more involved in supporting the validation of pharmaceutical applications.

The basic protocol for installing servers remains the same. Each individual server requires an installation and qualification protocol that inventories the hardware and software. Records of the serial numbers and internal components are required to create a baseline for both rebuilding the system in an emergency, and for future change control. The records should also include environmental controls, firmware revisions, service packs, and layered applications. The new disk controller technology for storage area networks will be required to follow the same set of rules as the protocols for installing a server.

Once the basic hardware and operating system level software is installed, the differences between single server applications and distributed applications begin to be more noticeable. Distributed applications may access databases on multiple servers, or the application data may be spread across a series of network-attached disks. The implementation of such an environment will affect backup, security, and disaster recovery. This is where the level of system management and operation team participation becomes crucial.

The most common event in distributed computing is for data to be entered or viewed via a user interface such as a network connected PC, thin client, Windows CE, or palm device. Once the method of entering and accessing data has been determined, the user interface needs to be examined to establish if validation of the client is required. The following questions must be asked: "Is the data captured or held on the client for any length of time? What happens to data that is entered on the client but does not complete and upload to the main database or processing application? Will another attempt to recover the lost data be made automatically by the application or must the data be re-entered and re-sent manually? Is Kerberos, DCE, or some other security tool storing information on the user device?" If the application is using any of these mechanisms or something similar, the client hardware and software should be included in the validation process.

## **Journal of Validation Technology**

If the data is being passed directly to a database or multiple databases, the structure of these databases must be included in the application's validation documentation. An installation script must also be crafted to test how the data is configured and where it is stored. Without this information the restoration of data in the event of an emergency will become an overwhelming task. As part of the validation process, both the application and data center teams should be involved in a test of the data recovery process.

A new twist on the validation process is the storage area network. The Storage Area Network (SAN) allows many servers to have disks placed on the network instead of directly connected to the server. Manufacturers of these devices are currently working on hardware controllers that will allow the disks to be shared by servers running the same operating system (and eventually by servers running different operating systems). The advantage to a SAN is that a disk will not be unavailable simply because a single server is down. A SAN allows many servers to access the same disk drive. When one server is down another server with access to the drive can provide communication with the required data. This decreases application unavailability to a very low level. The disadvantage of a SAN is the increased level of system management required to maintain and track data on the network disk farm. System management and application teams must share data to maintain a working validated SAN.

A SAN functions by allowing a disk storage controller to talk to multiple disks in a storage cabinet. The disk storage controller regulates traffic between the servers and disks on the network. The disk storage controllers need to be treated as servers when they are installed. Even if only one disk on a controller in the SAN contains validated data, all the controllers and disks should be included in the validation process. The assigning of disks to a server or application will need to be detailed in the installation and operational qualifications for the storage area network and the applications validation documents.

After the servers and distributed applications are installed, the procedures used to maintain the computing environment will need to be examined. Procedures that should be included in this examination are inventory, change control, backup and recovery, basic security, account, and password management.

The first set of procedures to be inspected cover inventory and change control. The inventory of the computing environment must be maintained and updated as hardware and software are added or removed from the computing environment. A suggested addition to the inventory would be a basic list of where information is stored on the disk farm. High-level information should include what disks are used for user files, application files, database files, log files and audit trails. Change control would remain essentially the same as before distributed applications were installed. Change control still needs to account for both planned and emergency changes. Additional data should be added to the change control system for tracking modifications made to the disk farm. If the top-level information for the disk drives distributed throughout the network is not managed, restoration of a failed drive may become impossible.

The next group of procedures to review includes backup and restore processes. Both backup and restore procedures require special attention when working with databases and other files that may be permanently opened by an active application. If databases are sharing data or synchronized with each other, the complexity of the backup and therefore the restore process will be increased. The best methods for backing up this type of application include shutting down the application and then performing a full backup of all the data files. Shutting down the application will close the data files and allow for a full backup of the files. This method of backup is sometimes referred to as a cold backup. The application may then be restarted after the backup is complete. This backup process may be too time consuming to perform on a daily basis. In cases where operations are 7x24 it is recommended that a periodic baseline cold backup be performed, weekly if possible, but no less than once a month. The application should also have a journaling and hot backup mechanism. Without this, the potential loss of data is very large. To improve recovery time, cold backups should be performed as often as possible. The application team and system management team should work together to determine the best method for managing backups of application data and database files.

When distributed data is restored, it must be handled in a very prescribed manner. The databases will need to all be restored or rolled back to the same point in their transaction history, or the records in the database will become corrupt and unusable. Tests of both backups and restores should be run on a periodic basis to confirm that the backups have executed as intended and that the data is recoverable. Remember, backup and restore are not disaster recovery but they are an integral part of disaster recovery and business continuity planning.

Another solution for backing up large, active database applications is duplicating the computing environment in an off site location. This may be done by building a remote cluster or using resources provided by the application. If a solution of this nature is built to limit downtime in the event of a site failure, the solution must be validated and maintained through change control with applicable procedures identical to the primary system. Periodically the master system or application should be taken off line and the recovery system should be tested to verify its functionality.

Along with testing the restorability of backups, disaster recovery should include the rebuilding of the entire validated environment at an off site facility. Disaster recovery plans should include a variety of topics including:

- Definition of off-site storage locations and the method of retrieval of the backup media.
- Means of reaching key personal
- Definition of meeting room locations
- Identification of the location of the data restoration exercise and the processes to be used
- List of end user contacts who may need to be contacted and the procedure for keeping this information current.

## **Journal of Validation Technology**

The final and most complicated section of managing servers in a distributed computing environment is security. Most of our servers today sit on a network that provides Internet access. Many first-time system managers rely solely on the firewall between the Internet and their network for security. This leads to a false sense of security. Firewalls only provide front door security. Each individual server should have operating system audit trails enabled. These audit trails should monitor failed and successful logins, failed and successful attempts to modify security files, password modifications, and attempts to access or modify controlled data. The operating system audit trails should also be augmented by additional application-level audit trails. Password management and account access restrictions should also be enabled and enforced to limit damage in the event of a security violation.

The operating system level audit trails will be necessary for monitoring the higher levels of a SAN disk farm. Vendors are currently supplying web-enabled software for the disk storage controllers in a SAN. This software should also be able to supply an audit trail. The details contained in the audit trail will vary from vendor to vendor and will be enhanced as the product matures.

As servers and disk farms multiply and increase in size, it will become increasingly difficult to individually review local audit trails. The solution to this dilemma is to purchase network applications that will monitor the audit trails and notify the system or security manager of any anomalies. These monitoring solutions are neither inexpensive nor simple to implement. The software does provide a mechanism to automate a task that people find tedious to perform and hard to complete without error.

The advent of distributed computing applications has a profound impact on the validation of the computing environment. No longer are applications restricted to a single server in a single location. Data is warehoused in multiple locations and is no longer defined by the system that hosts it. The location of data has become irrelevant to the end user; no longer do they need to know on what system their information is stored, they simply connect to the company network that grants access. While this has simplified life for the researcher and the manufacturer, it has drastically increased the complexity of managing the data and validating the application. No longer is validation a job limited to the individual responsible for the application. Validation has become a coordinated effort between applications development, systems management, data center operations, and the validation specialists. As applications and the network move to wireless technology, these groups will need to work more closely together to supply a secure, validated, and functional computing environment for the life sciences industry.