



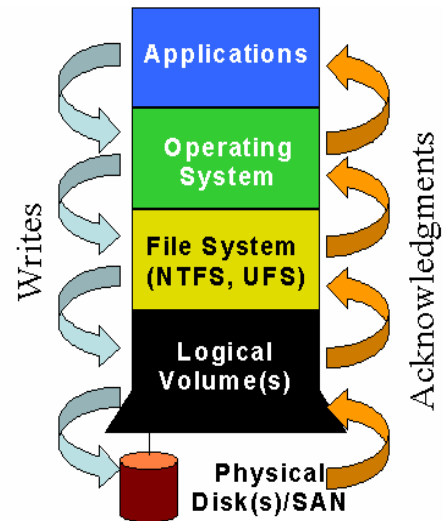
How does Double-Take[®] ensure Data Integrity

How does NSI[®] asynchronous replication technology ensure that data integrity is maintained, particularly with intra-dependent files like SQL and Oracle?

How Applications Write Data to Disk

All applications write data to persistent storage media such as a disk drive in three basic steps: Open/Create a file, Add/Update data to the file, and Close the file. Modern Operating Systems provide multiple software layers that provide assistance to the application write process to ensure that the file maintains a consistent state that the application can recognize the next time that it reads the file.

The Operating System Kernel provides a set of basic functions (API) that an application can use to read and write data to disk. This reduces the complexity by providing a single interface rather than a unique method for each possible type of application data or underlying media. The API provides transparency allowing applications to write to disk, tape, flash memory devices, etc. Most desktop applications store all of their data in memory and write an entire file to disk, relying on the File System to perform journal write protection to preserve data state. However, database applications must provide their own recovery features to preserve state.



Database applications, including SQL, Oracle[™], Exchange, and Notes[™] typically store much more data in their files than can fit into system memory. They open their files when they start, write transactions continuously, and then close their files only when they are shutdown. Databases ensure that all changes made to data are protected using a write method called “2-Phase Commit”.

The 2-Phase Commit method of writing data requires that the transaction be written to a disk-based log before any commitment is made to the client that the transaction was successful. The database application can replay the transaction log to update the database when it restarts, thus reducing the probability of data loss for transactions that were confirmed to the users. Once a transaction has been written to the database, it can be marked with a checkpoint which lets the database know that all of the instructions were successfully executed and the transaction can be removed from the log to make room for more new transactions.

Begin Transaction

```
Insert into Orders(CustName, OrderID)
values('John Doe', 4)
```

```
Insert into Items(OrderID, Product,
Price) values(4, 'Skis', $125.00)
```

```
Insert into Items(OrderID, Product,
Price) values(4, 'Poles', $15.00)
```

Commit

© 2005 NSI Software, Inc. All rights reserved. Double-Take[®], GeoCluster[®] and NSI[®] are registered trademarks of NSI Software, Inc. Balance[®] is a trademark of NSI Software, Inc. All other trademarks are properties of their respective companies.

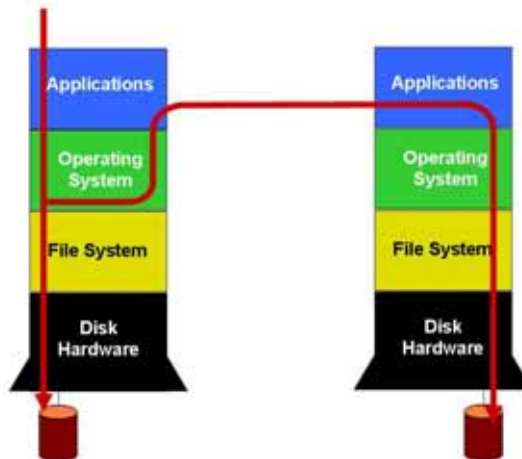
Microsoft, Windows Powered, Windows, Exchange, and SQL Server, are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Warning: No part of this document may be reproduced or transmitted in any form or by any means, electronic, or mechanical, for any reason, without the express written permission of NSI Software, Inc. The information in this document is subject to change without notice. Although we try to provide quality information, NSI makes no claims, promises or guarantees about the accuracy, completeness, or adequacy of the information contained in this document. Companies, names and data used in examples herein are hypothetical and/or fictitious unless otherwise stated.

How NSI Double-Take Protects Data Integrity

Double-Take® from NSI® Software integrates with the Windows Kernel I/O Sub-system which provides a serialized view of all I/O operations before they are passed to the File System Layer. This is a standard integration point where applications such as Anti-virus and Open File agents integrate and provides a fully consistent transactional view of all system writes.

Double-Take monitors the system I/O and replicates the write activity for user-defined volumes, directories or files. I/O operations that are exposed at this layer reflect the actual byte-level changes by the applications while they are writing to the files, so Double-Take replication does not require the database to be taken off-line or placed in a quiescent state before determining which changes have taken place. Double-Take also does not read data from the production system disks which. Other replication products that read data from disk negatively impact application scalability.



Double-Take protects application data integrity with a patented process that associates a unique sequential identifier for each logical write I/O operation that meets the volume, directory and name rules defined by the user; and stores a copy of the I/O in memory. When Double-Take receives an acknowledgement from the File System layer that the data was successfully written to disk then the copied I/O is given to the network sub-system for transmission to the target server using TCP/IP.

The TCP/IP network components encapsulate the Double-Take replication payload (transaction) in one or more packets and ensure that all associated packets are reassembled on the target system before presenting the complete I/O to the Double-Take software running on the target node. The Double-Take service inserts the transaction into the target I/O subsystem for write in the same order that the write operation occurred on the source. This creates an exact duplicate of the source data on the target server and fully preserves data integrity. Since Double-Take has maintained database state, then the the database is crash consistent. Following the application's standard soft recovery process, users can continue working.

How Double-Take failover works

So far, this paper has addressed how applications actually write data and validated the NSI approach to protecting it. The result so far is a disaster recovery or centralized backup solution for any transaction-based database, without the need for expensive application-specific agents.

The last piece of the puzzle is to provide near immediate availability of the database from the target server for high availability, via the failover technology within Double-Take or within the cluster solution of MSCS with GeoCluster® by NSI Software . Within a GeoCluster solution, NSI handles the replication as described above and the application availability is maintained by Microsoft Cluster Services. As such, that failover method will not be covered. Double-Take provides an alternative method for failover in situations where the application may not be "clusterable", where a client has chosen to not implement MSCS or the hardware may not support clustering. In addition, because deploying a cluster cannot be done "on top" of an existing server, many customers prefer the less obtrusive Double-Take approach of simply configuring a second server, the replication target, to fail over for the production server during an outage.

To do this, Double-Take provides up to five actions after determining that the production server has failed – including two script capabilities and three optional failover actions. The failover actions include assuming the production server's IP address, Machine name, and UNC file shares. For local file servers, this is completely automated and results in highly available file servers. For application servers, only one additional step would be required – the restarting of the application services, such as SQL or Oracle. One merely creates a PostFailOver script which starts the services, and the database soft recovery system handles everything else. Because it is a simple "batch file", any other tasks that are command driven can be inserted into either script for automation as part of the fail over process. The failback process is the reverse, with Pre- and Post- scripts plus the removal of the assume IP address, machine name, and UNC file shares.

There are other best practices on how to leverage these scripts and scenarios for when one would not fail over all of the aspects of the production server, but in summary Double-Take failover will:

1. **Pre Fail Over script** – ran before actual failover. Often clients will invoke a VSS snapshot or some other cleanup effort, email the helpdesk that the process has begun, or shut down non-essential services to free up CPU or Memory on the target server.
2. **IP Address(es) of Source** – allows for completely transparent failover. However, if the target server and source server are on different subnets, than this step should be disabled and additional networking will be handled. See NSI whitepaper on TCP/IP failover options.
3. **Machine Name of Source** – the target server will maintain its original identity but has the capability of assuming multiple other names.
4. **UNC file shares of Source** – this allows for transparent failover of file servers. Double-Take is intelligent enough to remap actual directories, such that if the Source used D:\ but the data was replicated to the Target's F:\ - the shares will be created correctly, including all relevant permissions and ACL's. Share information is updated to the target hourly.
5. **Post Fail Over script** – ran after steps 2-3-4 usually for the purpose of starting services such as SQL, Oracle, Notes or Exchange.

After the target has assumed the source identity, the dynamic nature of most Windows client applications will transparently and automatically reconnect to the server service, even if the client node had to re-resolve any DNS, WINS or authentication matters with the target server.

Closing Comment

NSI Software, maker of Double-Take and GeoCluster, is a Microsoft Gold partner whose products are consistently logo certified to the highest standard (Windows 2003 Server, Enterprise, and DataCenter). NSI has a long a partnership in supporting the Microsoft Windows server platform, critical business applications like SQL and Exchange, and advances in Windows storage.

NSI has over 55,000 licenses in production, including 13,000 on Exchange and 15,000 on SQL Server, making it the undisputed leader in protecting Microsoft environments through replication. Our technology is available through Dell, IBM, SunGard, HP and the traditional reseller channel.

We hope that you have found this information useful and please let your NSI representative know if you have further questions on this or other replication topics – or contact info@nsisoftware.com.