



# Advanced Server Protection Options



2 Hudson Place – suite 700  
Hoboken, NJ 07030

800-674-9495  
[www.nsisoftware.com](http://www.nsisoftware.com)

Powered by  **Double-Take**

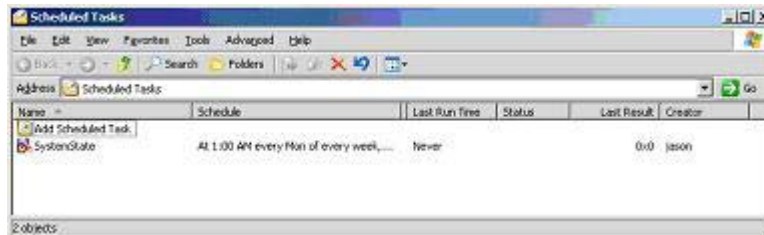
No part of this document may be reproduced or transmitted in any form or by any means, electronic, or mechanical, for any reason, without the express written permission of NSI Software. The information in this document is subject to change without notice.  
Companies, names and data used in examples herein are hypothetical and/or fictitious unless otherwise stated.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Due to the use of third-party tools described in this document, these concepts are offered to the reader for consideration only. These techniques are not supported by NSI service or support organizations – and should be tested for your environment before implementing into production.



First-time users of the backup utility should consider using the GUI to configure a backup of the system state (plus additional key files, such as INI's within program directories). During configuration of the backup job, one can schedule the job to run routinely - with a best practice being at least weekly.



The results will be an individual file (\*.BKF) instead of using actual tape or other media. By selecting the directory (where the backup file will reside) as part of the DoubleTake replication set, this BKF file will also be replicated to the target server. During any recovery, the BKF can be used to restore the system state (including registry and other in-use files).

By looking at the “scheduled tasks” applet, one will find the scheduled backup job is simply a command line of the ntbakup.exe utility. Before making changes, please consult the command-line reference guide or resource kit for your particular O/S. Below is a sample command line from a Microsoft Windows Server™ 2003:

```
ntbackup backup systemstate /J "System State Backup" /F
"s:\repset\SS_FS1.bkf" /D "System State of FS1" /V:yes /L:F
```

While several other command line options are available, the above script conducts the following:

ntbackup backup	Begin a Backup
systemstate	Include the System State (which in this case is the entire job).
/J "System State Backup"	Label the job for easier troubleshooting
/F "s:\RepSet\SS_FS1.bkf"	The target “media” is a file SS_FS1.bkf in the s:\repset directory. By ensuring that this directory is protected by DoubleTake, the backups will be sent to the target server.
/D "System State of FS1"	The internal NTBackup label for the job, which will be helpful during restoration.
/V:yes	Verify the integrity of the backup, by doing a second read and comparison after the backup is complete
/L:F	Enable “Full” logging, which is still minimal since the number of files is small.

As part of the recovery process, one should configure the new production server with the same O/S. Then, if the various system drive directories (e.g. Windows and Program Files) have been

replicated, those can be copied to the new server. If you are using a third-party backup package, one might consider backing up the remote source server's O/S volume (including system state) monthly. This will cause some network congestion, but once per month is typically tolerable. To restore, one would restore from tape, and then still replay the latest System State backup that was replicated via DoubleTake to the target server.

In either model, after the "new" server has a functioning O/S and application directory, then the only restoration is the data set (from DoubleTake), which will be seconds old. This results in near zero loss of data, including the precious registry information.

For more information about how to protect and restore the registry and other System State components, please visit the Microsoft website.

### **Capturing Specific Registry Hives**

Occasionally, one may need to secure only specific registry information – in order to correctly run the application from the target server during failover. For this purpose, the one can use Two Microsoft utilities = RegDMP and RegINI.

REGDMP (Registry Dump) is used to capture a registry hive (or individual key) to a text file. As an example, one can collect the NSI® Double-Take information by running:

```
REGDMP HKEY_LOCAL_MACHINE\Software\NSI Software\Double-Take\CurrentVersion >
nsi-dt.TXT
```

By piping the result to a text file, one now has all of the keys (and sub-hives) of DoubleTake. Again, using the native Task Scheduler within Windows, one can automate the REGDMP command to routinely dump registry hives to a text file (which would then be replicated to the target server).

REGINI (Registry Initialize) is used to insert registry values from a text file. Since the hive-location of where to insert is in the file, the command is simple:

```
REGINI nsi-dt.TXT
```

During failover, one would simply invoke a Pre-Fail-Over batch file (which runs after the need for failover has been determined, but before name/IP/shares are assumed). The batch file would apply the REGINI files to the registry of the target server, before any services were started (in the Post-Fail-Over script).

This registry technique has additional benefits, in that one can trim the file down to only those values which are non-default and then use it to "tune" servers. For example, for an enterprise deployment of Double-Take, one might instruct local personnel to do a default installation from

a master-CD and then click on a batch file (containing REGINI commands). The REGINI text file might include locations for page file or memory, activation codes or other tweaks.

It should be noted that any changes to the registry can have unforeseen effects and should only be done after adequate testing and understanding of results.

## **Dual Booted Failover**

In some one-to-one configurations (not many-to-one, without certain considerations), it may be desirable to not run from the target kernel's directory during failover. Instead, one may wish to have a complete source/production O/S on the target machine. This is also achievable using Microsoft's ability to boot multiple kernels.

Please recall that as a Windows server powers up, there is a brief menu that sometimes occurs referencing options such as:

- "Microsoft Windows 2000 Server"
- "Microsoft Windows 2000 Server [VGA Mode]"
- and perhaps other options.

These options are held in a BOOT.INI file, on the root of the first visible volume. A closer look at that file reveals that the selections determine where the O/S directory will be loaded from (e.g. C:\WINNT or C:\WINDOWS), as well as any optional switches.

By installing two kernels on the target platform, one might end up with a BOOT.INI like this one (with two kernel directories).

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS

[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Target - Microsoft Windows 2000 Server"
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Source - Microsoft Windows 2000 Server"
```

Normally, after 30 seconds (from the timeout value above), the C:\WINDOWS directory is used for the O/S kernel – and the server initializes as FSMT. However, in the right circumstance, one might want the target machine to be an identical O/S to the failed source/production server. In which case, one could:

- 1) install the target platform with the first O/S in directory C:\WINTGT
- 2) install the target platform with a second O/S in directory C:\WINNT (or whatever the name of the directory is on the actual source server).
- 3) Ensure that the BOOT.INI default is to the target O/S.
- 4) Install Double-Take on both servers and replicate the source O/S to the target.

This maintains all settings and drivers to a second O/S directory on the target, without changing the target's native directories at all.

Then, to fail over the target server, one does not select the default fail over of Name, or IP, or Shares. Instead, in the Pre-Fail-Over batch file:

- Replace the normal BOOT.INI file with one that is exactly the same, but points to the alternate O/S kernel directory.
- Run SHUTDOWN.EXE /R (another Microsoft utility) with the “reboot” option.

With the boot.ini file changed, the server will power down and then on power-up become the actual failed source server. Obviously hardware should be near identical and failover of multiple sources simultaneously is not possible. But for the right application server, this is a valid configuration.

During Failback, there are two options for restoring one's data:

- 1) One can again replace the BOOT.INI file to the original (which points to the target) and then reboot the target platform. Then, after bringing the real source server back on-line, one can use the DoubleTake Restoration Manager to push the data back to the source server. This is same as if one had done a traditional failover, and can use all existing NSI documentation.
- 2) One can dual-boot the source (using the same methodology), whereby the source platform might come up as FS1 or FS1REAL.
  - By booting as FS1REAL, the users can continue to run from the target platform in its failed-over state as FS1.
  - One can configure DoubleTake on the target platform to replicate its data to FS1REAL (while the users are still on the target-FS1). This can even be done in advance, so that when FS1REAL is first powered up, DoubleTake uses its normal auto-reconnect and auto-remirror functions to get the machines back in sync.
  - After FS1REAL has completed its mirror from the target server, then both machines have the same data.
  - So, “Failback” is simply failing-back the target platform (which removes the FS1 identity from the network) and then powering up the source platform as FS1. Since the mirroring had already been done, this can result in a total failback window of under 10 minutes, with no restoration step required.

The dual-boot model is not supported through the NSI Technical Support organization, unless the implementation was done via an NSI Professional Services engagement. This is due to the numerous complexities and unknowns from a support perspective. However, best effort support will still be done past the O/S issues.

Dual-Booting the Source machine can provide for faster failback, by not requiring the restoration of data while users are off-line.

Dual-Booting the Target machine can provide for a completely separate kernel to be run during failover, which also allows for registry and DLL's from the source to be isolated.

### **Virtual Machines (e.g. VMware or Connectix)**

One more alternative that is available for advanced failover methods is the use of virtual machines. Virtual machines act similar to “virtual servers” inside of a cluster, except that they are even more autonomous.

Virtual machines run in a window (as a separate process and separate memory space) on a physical server. The physical server is often called a “Host OS”, while the virtual server is a “Guest OS”. Using these technologies, in conjunction with NSI DoubleTake, offers failover options - all of which are described in more detail on a separate whitepaper on NSI's website. The short solution descriptions are as follows:

- 1) If the source server is running VMware (and the production server is actually in the GuestOS on the source hardware), then the entire production server O/S is held in a physical file. By running DoubleTake on the source host OS, it can replicate that file to the target server. This is analogous to taking the actual hard drive of a server and running it on another platform during a crisis.

The failover of this solution is to simply start the GuestOS on the target server. Every driver and DLL of the entire virtual C:\ of the production server is contained within the physical file. And the production O/S and applications will transparently move and failover.

Other options are possible when running VMware on the target only.

- 2) If the source server(s) are configured to replicate to the target's GuestOS, then one can create many “one-to-one” solutions within one set of target hardware. A popular use of this is by third-party off-site or storage-provider companies. In these cases, an SSP can utilize one physical server across multiple customers – because each customer target is a GuestOS (which is completely isolated and secure). In this configuration, the target

HostOS becomes irrelevant and the solution is a typical DoubleTake configuration; except that the hardware is shared.

- 3) If the source server(s) are configured to replicate to the target's HostOS, then one can configure failover to simply power up a GuestOS and mount those replicated data areas. This provides for a failover of multiple failed sources, simultaneously. It also provides for a single target hardware platform to fail over multiple (and perhaps incompatible) applications, because the applications run within independent Guest OS's.

More details on using VMware (and other virtual server technologies) is available on the NSI website.

For all of these solutions, it is recommended that you consult with NSI Software's Professional Services group (or an authorized NSI partner) for these advanced configurations.

All of these solutions are based around NSI's fundamental philosophy that all business continuity efforts start with protecting the data. From there, it is simply a matter of what you want to do with it.

**NSI Software** knows how to protect applications running on Windows file systems. "Business Continuity through Replication" is the single focus of every person in our company. That focus, and the quality of our products, has helped NSI forge relationships with HP®, IBM®, Dell®, SunGard®, Microsoft® and probably your preferred reseller-integrator.



For over 10 years, NSI has been providing the products, services, and support to help you be successful in protecting your most critical applications...

We'd like the chance to prove it to you.

© 2003 NSI Software, Inc. All rights reserved.

Double-Take®, NSI® and GeoCluster® are registered trademarks of NSI Software, Inc., Balance™ is a trademark of NSI Software, Inc.. and all are used with permission of the trademark owner. 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.

Our Vision

*To be the leading provider of data protection & high availability software solutions for 24x7 business operations*

Our Offer to You

*We would like to become your partner in ensuring the continuous operations of your business.*

*Please allow us the opportunity to talk to you about your specific data protection needs and to discuss our products and services that may apply.*

*Products that Protect your Data*

*Services that Ensure your Success*

For more information on NSI's products and services, please contact NSI.

NSI Software, Inc. - Corporate Office

Two Hudson Plaza, Suite 700  
Hoboken, NJ 07030  
800-775-4674 or 201-656-2121  
Fax: 201-656-2727



NSI Software, Inc. – Inside Sales

8470 Allison Pointe Blvd. Suite 300  
Indianapolis, IN 46250  
800-674-9495  
Fax: 317-598-0187

Or visit us on the web at [WWW.NSISOFTWARE.COM](http://WWW.NSISOFTWARE.COM)

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. The information in this document is subject to change without notice. Companies, names and data used in examples herein are hypothetical and/or fictitious unless otherwise stated.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Due to the use of third-party tools described in this document, these concepts are offered to the reader for consideration only. These techniques are not supported by NSI service or support organizations – and should be tested for your environment before implementing into production.