Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2009 Hewlett-Packard Development Company, L.P.
Contents

SQL Server Storage Architecture ........................................... 5
  Overview ........................................................................... 5
General SQL Server Storage Requirements ............................... 5
  Base SQL Server Software ................................................. 6
  Master Database .................................................................. 6
  tempdb Database ............................................................... 6
  Model Database .................................................................. 7
  User Databases .................................................................. 7
  Log Files ........................................................................... 7
  Backup Files ...................................................................... 8
Characterizing Application Workloads ...................................... 8
  Online Transaction Processing (OLTP) ................................ 8
  Online Analytical Processing (OLAP) ................................. 8
Impacts of Database Operations ............................................... 8
LeftHand Networks Integration with SQL Server ...................... 10
  Thin Provisioning for Storage Efficiency ............................ 10
  SmartClones ..................................................................... 10
  Quiesced Snapshots with the LeftHand VSS Provider .......... 11
SAN Best Practices for SQL Server .......................................... 13
  Performance Recommendations ....................................... 13
  Using Windows Performance Monitor to Measure SQL Server Performance ........................................ 13
  Setting up Windows Performance Monitor ......................... 14
  Saving A Performance Monitor Log for Analysis ................. 15
  Volume Configuration, Partition Alignment ......................... 19
    Volume Configuration ...................................................... 19
    Partition Alignment ....................................................... 20
    Aligning a partition with diskpart for Windows Server 2003 ......................................................... 20
  Backup Config and Operation ............................................. 21
Overview

Microsoft® SQL® Server is today’s fastest growing database environment with over 75% of enterprises using the product. With SQL Server 2005, Microsoft has established SQL Server as a major application infrastructure for enterprise class applications and the new features of SQL Server 2008 promise to accelerate its adoption. SQL Server’s roots are in the plethora of back room and homegrown applications that all companies have, but now SQL Server is also running the type of business-critical applications that were the exclusive domain of Oracle and DB2. Storage plays a key role in any major database environment. Reliability, availability, data protection, scalability, management, and security all depend on the underlying storage. SQL Server must reside on a storage subsystem that delivers enterprise-class storage management to meet the ever-increasing business requirements. A recent InfoPro survey of SQL Server database administrators indicated that storage design, provisioning and administration consumed nearly one third of a their time. LeftHand SANs designed to simplify operations, improve utilization and cost-effectively scale as your needs grow over time. Storage virtualization removes the complexity of volume and LUN design. Thin Provisioning removes the prerequisite of complex storage calculations and imprecise predictions. Integrated snapshots and SmartClones optimize backup and testing operations. Network RAID and multisite capabilities give administrators powerful and simple new options for high availability.

General SQL Server Storage Requirements

Properly configuring the storage of a SQL Server instance can significantly improve the overall performance of the system. While there are several common configurations that should usually be followed (e.g. grouping similar I/O patterns on the same volumes), the majority of the storage configurations will be dictated by the I/O characteristics of the applications accessing the
databases. A solid storage configuration plan should begin with a thorough review of the applications specific needs. The recommendations covered in this paper are simply 'recommendations' and should be overridden by the specific requirements of each unique application and situation.

Additionally, avoiding unnecessary storage complications will better serve the environment and ease any future maintenance of the system. A simple storage design will provide a combination of performance and flexibility that will work better into the overall solution rather than a complicated system that rapidly becomes unmanageable. A good rule of thumb is to simplify where possible and to dedicate resources when performance or maintenance requirements dictate.

The following sections cover some of the basic storage requirements and characteristics of a SQL Server instance.

**Base SQL Server Software**

The storage requirements for a SQL Server instance vary depending on the software version, computer type (32 vs. 64 bit), add-on components, and any optional server tools installed. At least 2 GB of storage should be set aside for the SQL Server executables. Refer to Microsoft’s specific storage requirements for the version of SQL Server.

**Master Database**

The Master database stores system level information on the entire SQL Server instance including system configurations, account and specific database information. The Master database is approximately 6 MB after an initial installation and grows as databases are added to the SQL Server instance. This database is commonly placed on the same volume as the SQL Server executables, but it can be separated out onto the SAN depending on the specifics of the environment.

**tempdb Database**

SQL Server uses a single tempdb database to store the temporary database objects including user generated temporary tables, sort operations, stored procedures, row sets for index rebuilding, and cursors. By default, the tempdb is set to 8 MB with autogrow enabled. The tempdb resets to its original size when SQL Server is restarted and should be increased if it routinely grows beyond its current set size.
Properly pre-sizing the tempdb to match the SQL Server and application workload can significantly improve the overall system performance. Refer to Microsoft’s Working with tempdb in SQL Server whitepaper for details on sizing the tempdb. The tempdb storage volume is an ideal candidate for LeftHand’s Thin Provisioning detailed later in this paper due the difficulties in predicting of the size of the tempdb.

For SQL Server instances with a high utilization of tempdb, increasing the number of data files to equal the number of CPU’s (counting dual and quad-core as two or four CPU’s respectively) can avoid latch contention on allocation pages.

Model Database

SQL Server uses the Model Database as a template for new user created databases. The default Model Database is 3 MB in size and can stored on the same volume as the executables unless significant changes are made to it.

User Databases

User databases store the data of each user created database schema in the SQL Server instance and are created by copying the Model Database. Their size and configuration should be created to meet the requirements and usage model of the applications that access them. User databases can be split into multiple, equally sized data files at a ratio of 1 data file per 4 CPU’s present (again, counting dual and quad-core processors as two or four respectively) to avoid latch contention. User databases are ideal candidates for Thin Provisioning to maximize storage utilization on the SAN.

Depending on their usage, user databases with similar I/O patterns can be grouped together on common volumes. Databases with higher utilizations or specific back-up and recovery requirements should be placed on their own dedicated volumes. Again, simplify where possible and dedicate resources where performance or maintenance requirements dictate.

Log Files

Each database in the SQL Server instance has its own log file. As a rule, SQL Server should have at least one storage volume dedicated to log files to separate their sequential I/O from the random I/O that most user databases generate. Logs for databases with a higher utilizations or specific back-up and recovery requirements should be placed on their own dedicated volumes.
Backup Files

The size of SQL Server backup files varies depending on the type of backup being performed (full, full differential, log, file, etc.) and the amount of DML since the previous backup. Ideally, a second SAN should be used to store the backups with databases grouped into volumes based upon their specific RPO and RTO requirements.

Characterizing Application Workloads

Understanding the specific I/O characteristics and requirements of an application can have a significant impact on the overall storage design. This section details some of the terminology used to describe SQL Server I/O patterns and their impacts on the storage design.

Online Transaction Processing (OLTP)

OLTP applications generally have a high number of concurrent users and generate a high level of read, write, update, delete, and create transactions. OLTP applications are driven by the amount of random I/O generated and should be optimized to handle the burst nature of the applications accessing them. Systems should be scaled to meet performance requirements of their peak workload.

Online Analytical Processing (OLAP)

OLAP applications are generally characterized by a higher-level read activity associated with demanding and complex queries from a smaller number of concurrent end user connections. This workload type is commonly generated by analytical or Decision Support Systems. OLAP systems should be tuned for sequential reads and writes with extra attention given to the performance of the tempdb.

Impacts of Database Operations

Several internal database operations should be taken into account while characterizing the database workload as they can have an impact on the storage subsystem. Administrators should include them in the peak workload calculations depending on the frequency and length of the operation.
Backup

Standard database backups perform large sequential reads of the data and log files as well as large sequential writes to the backup disk.

Bulk loads

Bulk loads perform large sequential writes to the data and log files.

Checkdb

Checkdb verifies the allocation and structural integrity of all the objects in the specified database by executing sequential reads of the data files.

Checkpoints

Checkpoints write dirty pages (data pages that are in the buffer cache that have not been written to disk) to disk to enable faster recovery and to allow for efficient reuse of log space.

Index creation

Index creation performs sequential writes to the data and log files, random reads and writes in the tempdb and random reads and writes to the database if the sort result does not fit in memory.

Shrink/Defrag

Performs random reads and writes to both the data and log files.
Thin Provisioning for Storage Efficiency

SAN/iQ software is structured so that all storage blocks are allocated to volumes the first time they are written to by an application, with optimizations that allow blocks to be allocated without impacting write performance. Allocating blocks on demand allows the software to stripe and replicate blocks across the storage cluster in the best way possible at the time that blocks are allocated to a volume.

Thin provisioning dictates whether or not a volume’s blocks are reserved ahead of time. A fully provisioned volume has its blocks reserved ahead of time, while a thin-provisioned volume has none. This makes it easy to change a volume back and forth between full and thin provisioning: what changes underneath is the storage cluster’s accounting of how many blocks it has available, not the actual allocation of blocks which is always done on demand.

User databases are prime candidates for thin provisioning as they contain the majority of the data associated with a given database and are usually not sequential. Thin provisioning removes the necessity of accurately predicting future database size by allowing administrators to more generously provisioning volumes without wasting resources within the SAN. SAN/iQ volumes can also be resized without interruption for an added level of comfort.

SmartClones

SAN/iQ SmartClone software is built using the space-efficient SAN/iQ thin-provisioning architecture. This means that volume clones can be used without limitation along with all other SAN/iQ features like Network RAID, Thin Provisioning, Snapshot, and Remote Copy. The feature works by taking any volume or snapshot and making one or many clones in an instant. The cloning function makes a permanent, read/write volume on the SAN. This eliminates any requirement for a volume, snapshot, or any of its predecessors.
to exist after the clone has been created. The clone shares the original volume's blocks and uses copy-on-write semantics. There is never a need to run de-duplication software because blocks are never duplicated to begin with.

For SQL Server, SmartClones are a fast and space efficient method for rolling out new database instances. For example, creating a copy of a 1 TB database for upgrade testing could take a fair amount of time not to mention requiring a full TB of storage. SmartClones are created instantaneously and are thin provisioned allowing for rapid testing and rollout of new instances.

**Quiesced Snapshots with the LeftHand VSS Provider**

In a Microsoft Windows 2003 and 2008 Server environments, Microsoft delivers a software Volume Shadow Copy Service (VSS) framework facilitating communication between applications and storage allowing consistent point in time copies of data (Shadow Copies) for archiving and restoration. The VSS framework consists of requestors, writers, and providers. Requestors initiate and manage the backup, and writers prepare an application or file system for shadow copy creation. The LeftHand VSS Hardware Provider interfaces between the VSS Framework and the storage, performing the snapshot within the storage hardware without operating system intervention.

The SAN/iQ VSS Provider simplifies the process of using LeftHand snapshots with SQL Server and third party backup software. This technology allows SAN/iQ to create consistent point-in-time copies of critical application data that can easily be used to recover database instances. These snapshots are highly reliable as they are created after the application is quiesced; are space efficient using SAN/iQ's Thin Provisioning features; and have less impact on the SQL Server services than traditional snapshots as they are created on the back-end SAN rather than on the database server. Creating snapshots of both the data and log volumes simultaneously creates a fully coverable image of the database that can be mounted to any SQL Server instance.

Microsoft’s VSS Developer’s Kit contains the vshadow utility that will function as a minimal requestor (backup application) to initiate the creation of a shadow copy. Snapshots may remain local (non-transportable) or may be transported to another server for performing the backup. This utility can be used as part of a full back-up procedure, as a transportable, one-off back-up creator, or for testing and verification purposes as done here.

1. Download and install the **Volume Shadow Copy Service SDK 7.2**.
2 Install the SAN/iQ VSS provider and Authentication Console from the SAN/iQ Solution Pack for Microsoft Windows Server installation package.

3 Configure the SAN volumes and Microsoft iSCSI initiator connections as normal.

4 Configure the LeftHand VSS Hardware Provider in the Authentication Console as detailed in the SAN/iQ Solution Pack for Microsoft Windows User Guide.

5 Create the VSS snapshot of the volumes (represented as the e: and f: drives in this example) from a DOS prompt with the following:

   vshadow -p e: f:

6 The snapshot creation can be verified in the SAN/iQ Centralized Management Console (CMC) as shown here.

   ![SAN/iQ Centralized Management Console (CMC) screenshot]

The resulting snapshots can either be rolled back in the SAN or directly mounted to a SQL Server instance as follows:

   CREATE DATABASE <new database name> ON
   (FILENAME = '<path to database .mdf data file>'),
   (FILENAME = '<path to database .ldf log file>')</n   FOR ATTACH

In addition, the Windows Server 2008 Backup utility is also VSS-based, and is designed for production use of snapshots with Microsoft applications. See the link in the references section for more information.
SAN Best Practices for SQL Server

Performance Recommendations

Performance recommendations for SQL Server vary depending on the workload type and concurrency of the database. Windows Performance Monitor can be used to measure the existing disk usage as a baseline. The simple mantra of ‘monitor, tune, repeat’ should be followed.

Using Windows Performance Monitor to Measure SQL Server Performance

The preferred method to measure application server performance connected to the SAN is to use Windows Performance Monitor (perfmon.exe) and sample the appropriate PhysicalDisk counters for the SAN volume(s) in question. The following are relevant performance counters and what they measure:

<table>
<thead>
<tr>
<th>Performance Counter</th>
<th>SAN Measurement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Disk Sec/Read</td>
<td>Read I/O Latency of the volume</td>
<td>Measured in Seconds. Typical values are in milliseconds.</td>
</tr>
<tr>
<td>Avg. Disk Sec/Transfer</td>
<td>I/O (Read &amp; Write) Latency of the volume</td>
<td>Measured in Seconds. Typical values are in milliseconds.</td>
</tr>
<tr>
<td>Avg. Disk Sec/Write</td>
<td>Write I/O Latency of the volume</td>
<td>Measured in Seconds. Typical values are in milliseconds.</td>
</tr>
<tr>
<td>Avg. Disk Queue Length</td>
<td>Average # of pending I/O requests for the volume</td>
<td>Measured as the raw number.</td>
</tr>
<tr>
<td>Current Disk Queue Length</td>
<td>Current # of pending I/O requests for the volume</td>
<td>Measured as the raw number.</td>
</tr>
<tr>
<td>Disk Bytes/sec</td>
<td>Total data throughput for the volume</td>
<td>Measured in Bytes/sec. Typical values are in Megabytes/sec.</td>
</tr>
<tr>
<td>Disk Read Bytes/sec</td>
<td>Read data throughput for the volume</td>
<td>Measured in Bytes/sec. Typical values are in Megabytes/sec.</td>
</tr>
<tr>
<td>Disk Write Bytes/sec</td>
<td>Write data throughput for the volume</td>
<td>Measured in Bytes/sec. Typical values are in Megabytes/sec.</td>
</tr>
</tbody>
</table>
Setting up Windows Performance Monitor

1. Open Windows Performance Monitor (Start Menu>Accessories>System Tools>Performance Monitor, or run the perfmon.exe program).
2. Click the X icon several times to remove any existing counters.
3. Click the + icon or right-click on the graph area to open the Add Counters function.
4. Configure the Add Counters window as shown below. Make sure to include all counters listed above. (To simplify the process, choose “All counters” from the column on the left to avoid having to individually select counters). Add the PhysicalDisk counters listed in the table above, and then click the Add button.
5. Make sure to choose the proper physical disks from the section on the right. Unless directed to do otherwise, select all disk instances and do not choose the _Total. Selecting all the individual disks allows for a more detailed analysis of the performance data.
6. Select Close. *Choosing the “view report” icon rather than the “view graph” icon will generally reveal more useable results.

<table>
<thead>
<tr>
<th>Performance Counter</th>
<th>SAN Measurement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Reads/sec</td>
<td>Read IOPS (I/Os / sec) for the volume</td>
<td>Measured as the raw number.</td>
</tr>
<tr>
<td>Disk Writes/sec</td>
<td>Write IOPS (I/Os / sec) for the volume</td>
<td>Measured as the raw number.</td>
</tr>
<tr>
<td>Disk Transfers/sec</td>
<td>Total IOPS (I/Os / sec) for the volume</td>
<td>Measured as the raw number.</td>
</tr>
</tbody>
</table>
Saving A Performance Monitor Log for Analysis

It is common for Administrators to monitor performance on a longer-run basis. If needed, setup a performance counter log by expanding Performance Logs and Alerts, right-click Counter Logs, and select New Log Settings.
Provide a meaningful name and choose to manually create the data collector set.

Choose to create a data log of Performance counters.
Click “Add…” to add counters to the collection set. The sample interval (rate) can also be changed here. The default of 15 seconds is usually suggested.

Choose the counters that you wish to add. If unsure, choose all physical disk counters and all instances.
Choose where the data (logs) are to be saved. It is a good idea to save the logs to a location that is not being monitored for performance, as to minimize the effect of the logging itself on the data.

Choose to save and close the Data Collector Set.
Volume Configuration, Partition Alignment

Volume Configuration

SQL Server requires a number of volumes to be set up. Even though it would be possible in a single server environment to specify a single iSCSI volume for everything in SQL Server, in production environments it makes sense to create multiple volumes for various databases. Some general guidelines are:

- Keep all the main databases in their own volumes to simplify their high availability and backup requirements.
- Put database logs in a separate volume from the databases as they have distinct IO patterns.
- Use thin provisioning for the database volumes.
- Use standard provisioning for the log volumes.

The output file can be saved as a binary log (blg) or a comma-separated file (csv). To change the output format, select the Log Files tab. The .blg format file can be viewed later in Performance Monitor. The .csv file can be imported into a spreadsheet or database for further analysis.
Partition Alignment

This section describes how to configure Windows Server 2003 disk partitions to be aligned optimally for LeftHand storage. Windows Server 2003 default partition set does not align the partition to the physical disk that the partition is on. Correctly aligning the partition helps reduce latency when writing to the partition by eliminating unnecessary disk writes and reads that occur when partitions are not aligned. Windows partitions should be aligned at 64K for best results.

With a physical disk that maintains 64 sectors per track, Microsoft Windows always creates the partition starting at the sixty-fourth sector, therefore misaligning it with the underlying physical disk. To be certain of disk alignment, use Diskpart.exe, a disk partition tool. Diskpart.exe is a tool provided by Microsoft in the Windows Server 2003 Service Pack 1 support tools that can explicitly set the starting offset in the master boot record (MBR). Setting the starting offset correctly will align Exchange I/O with storage track boundaries and improve disk performance. Microsoft Exchange Server 2007 writes data in multiples of 8-kilobyte (KB) I/O operations, and I/O operation to a database can be from 8 KB to 1 megabyte (MB). Therefore, make sure that the starting offset is a multiple of 8 KB. Failure to do so may cause a single I/O operation spanning two tracks, causing performance degradation.

**NOTE:** this can only be done when creating a new partition before formatting. It is not possible to align a partition that has data on it already without losing that data.

Aligning a partition with diskpart for Windows Server 2003

1. Open a command prompt and type diskpart.exe
2. Enter > list disk
3. Note the disk number that you want to create the partition on.
4. Enter > select disk (disk number)
5. Enter > create partition primary align=64
6. Enter > assign letter (the letter you want the drive to have)
   - Or enter > assign mount (the path of an empty dir to mount the drive to)
7. Enter > exit (to exit diskpart)
Backup Config and Operation

SQL Server Backups

The embedded SQL Server backup mechanism can be used. The benefit here is that it is a consistent procedure with other SQL Server databases being administered. However, the administrator must know which databases to back up and more importantly, how to restore them. SQL Server backups support disk only and are designed to be used in conjunction with a tape backup product.

Continuous Data Protection Products

Continuous Data Protection (CDP) products monitor all I/Os for a particular application and copy these I/Os to a CDP server which then applies them to a duplicate information store on that server. For databases like SQL Server, they generally use VSS as a way to synchronize the database periodically to provide consistent “recovery points” that administrators can restore back to. The benefits are that they provide very little loss of data in the event of a restore, they offer granular restore of individual items and they offer point-in-time recovery of data in case of corruption. These products understand the relationships between all the pieces for supported applications like Exchange and SharePoint and coordinate getting all those pieces back in the proper place, thus eliminating the expertise required to do restores from snapshots. Most CDP products also have a “remote” capability to ship data to another CDP server at a remote location.

The cons of these products are that they generate large amounts of LAN traffic as all the backup data is transferred “live” over the LAN to the CDP server. The resulting capacity requirements can also be quite large, ranging from three to seven times the size of the production data. Even though the backups are “continuous” and therefore do not require a backup windows on the application servers, restores are done over the LAN by copying data and can therefore become long if the amount of data being recovered is large. CDP products have become a good method for providing the ability to do logical restore of individual items, but not as a good mechanism for full restores of entire systems.

Microsoft’s System Center Data Protection Manager (DPM) is a good choice here. Other vendors in this space are Asempra, FilesX, AppAssure and TimeSpring from Doubletake Software. Make sure that the vendor supports SQL Server 2005 and 2008.
Recommendations

- Use nRAID Level 2 for all volumes containing SQL Server databases and/or files. This protects against double drive failures, any hardware failure within a storage node and any SAN network failures.

- Use a CDP product that supports SQL Server as the primary backup. Microsoft’s Data Protection Manager provides item-level restore capability and full point-in-time recovery of the SQL Server environment in the event of a total failure of the production SAN or corruption of any of the primary databases.

- Use LeftHand’s Multi-site SAN configuration to spread your SAN across two data centers for protection against local site or data center disasters. This is included with SAN/iQ and does not require any additional administrative expertise or hardware.

- Use the remote copy facility of your CDP product for protection against metro-wide disasters. This provides for a long-distance remote copy and takes advantage of those product’s recovery capabilities. LeftHand’s Remote Copy facility can also be used here, but the recovery management capabilities are not as complete.

- Taking VSS-snapshots of your database files and log volumes of the SQL Server databases on a nightly basis provides a second tier of protection.

- Use a tape-based backup solution for a third tier of protection or long-term archive.

nRAID Level 2 combined with a CDP product like Microsoft’s System Center Data Protection Manager provide a complete solution for protecting your SQL Server environment from all but large-scale disasters and providing for easy recovery of logical items such as sites, site collections and documents.

High Availability

LeftHand’s highly available SANs complement many of the SQL Server high availability features. This section provides a brief overview of several techniques and the best practices for extend configuring the SAN.
Failover clustering

Microsoft Cluster Service (MSCS) combines two or more server nodes into a single resource group that provide failover should one server become unavailable. The SQL Server instance appears as a single server on the network aliased by the fail over cluster instance name. Shared storage is key requirement for failover clustering.

LeftHand’s Multi-site SAN functionality compliments failover clustering by creating a single storage node that spans two or more locations. The SAN volumes simply remain live in the event of single site failure. Additionally, the SAN/iQ Multi-site functionality resynchronizes the SAN nodes with the system is restored.
Additional Resources

For the latest information about Microsoft SQL Server 2005, see the SQL Server site: http://www.microsoft.com/sql/default.mspx


Appendix

**Moving tempdb**

This script can be modified with the directory path to move the tempdb after the initial installation of SQL Server. SQL Server must be restarted to complete this change.

```
USE master
GO
ALTERT DATABASE tempdb
    MODIFY FILE (NAME = tempdev, FILENAME = "\[New data folder path]\tempdb.mdf")
GO
ALTERT DATABASE tempdb
    MODIFY FILE (NAME = templog, FILENAME = "\[New log folder path]\templog.ldf")
GO
```

**Adding and Sizing tempdb Data Files**

This script can be modified with the directory path to size and split the tempdb. SQL Server must be restarted to complete this change.

```
USE master
GO
ALTERT DATABASE [tempdb]
    MODIFY FILE ( NAME = N'tempdev', SIZE = [Calculated]KB)
GO
ALTERT DATABASE [tempdb]
    ADD FILE (NAME = N'tempdev_02',
              FILENAME = N'[Second data file path]\tempdev_02.ndf',
              SIZE = [Calculated] KB, FILEGROWTH = 10%)
GO
ALTERT DATABASE [tempdb]
    ADD FILE (NAME = N'tempdev_03',
              FILENAME = N'[Third data file path]\tempdev_03.ndf',
              SIZE = [Calculated] KB, FILEGROWTH = 10%)
GO
ALTERT DATABASE [tempdb]
    ADD FILE (NAME = N'tempdev_0N',
              FILENAME = N'[Nth data file path]\tempdev_0N.ndf',
              SIZE = [Calculated] KB, FILEGROWTH = 10%)
GO
```