Environment Setup and Installation

Standard Operating Procedure - Enterprise Team Foundation Server

# Service Contacts

|  |  |
| --- | --- |
| Name | Role |
| Char Moser | IT Service Manager |
| Heather Bufkins | 3M Database DBA |
| Emily Gregerson | IT Operations Manager |
| Mike O’Brien | ETFS Service Manager |

# Procure Machines

Enterprise TFS is run on virtual front end Application tier machines. New virtual machines can be requested via the 3M Server Express website at <http://serverexpress/default.aspx>. Data tier machines are a mix of physical and virtual, and are managed by the 3M Database team. Requests to add, modify, or remove should be requested through the Database team.

# Initial OS Configuration and Regular Management Tasks

* Collect a performance counter baseline for a representative period of time to identify any bottlenecks and serve as a useful diagnostics tool in the future.  A collection over a 24 hour period on a weekday @ 1-5min intervals to a local file should be sufficient. Don’t know which counters to collect? Download the [PAL tool](http://pal.codeplex.com/) and look at the “threshold files” for “System Overview” on all your servers, “SQL Server” on your data tier servers, and "IIS" and ".NET (ASP.NET)" for your application tier servers.
* Ensure antivirus exclusions are correct for TFS, SQL and SharePoint. ([KB2636507](http://support.microsoft.com/kb/2636507))
* Ensure firewall rules are correct. I had an outage once where the network profile changed from ‘domain’ to ‘public’ due to a switch gateway change, and our firewall policy blocked SQL access for the ‘public’ profile which effectively took SQL offline for TFS.
* Ensure page file settings are configured for an appropriately sized disk & memory dump settings are configured for Complete memory dump. If you get a bluescreen, having a dump greatly increases your chances of getting a root cause + fix. ([KB254649](http://support.microsoft.com/kb/254649)), test the settings using [NotMyFault.exe](http://support.microsoft.com/kb/969028) (during a maintenance window, of course)
* Don’t run SQL or TFS as a local administrator.

# Initial SQL Configuration

* [SQL Pre-Deployment Best Practices](http://technet.microsoft.com/library/Cc966412) (SQLIO/IOmeter to benchmark storage performance)
* SQL recommended IO configuration. SQLCAT [Storage Top 10 best practices](http://technet.microsoft.com/en-US/library/cc966534)
* [Check disk partition alignments](http://technet.microsoft.com/en-us/library/dd758814(v=SQL.100).aspx) for a potential 30% IO performance improvement (especially if your disks were ever attached to a server running Windows Server 2003, but sometimes if you used pre-partitioned disks from OEM)
* Ensure that [Instant File Initialization is enabled](http://www.sqlskills.com/blogs/kimberly/instant-initialization-what-why-and-how/) (if the performance vs. security trade-off is appropriate in your environment. The article has more details). This enables SQL to create data files without having to zero-out the contents, which makes it “instant”. This requires the service account that SQL runs as to have the ‘Perform Volume Maintenance Tasks’ (SE\_MANAGE\_VOLUME) permission.
* Separate LUNs for data/log/tempdb/system.
* [Multiple data files for TempDB](http://blogs.msdn.com/b/chandrur/archive/2006/01/10/configuring-tempdb-for-optimal-tfs-performance.aspx) and TPC databases. ([See here for guidance](http://www.sqlskills.com/blogs/paul/a-sql-server-dba-myth-a-day-1230-tempdb-should-always-have-one-data-file-per-processor-core/) on the “right” number of files. If you have less than 8 cores, use #files = #cores. If you have more than 8 cores, use 8 files and if you’re seeing in-memory contention, add 4 more files at a time.)
* Consider splitting tbl\_Content out to a separate filegroup so that it can be managed differently
* [Consider changing ‘max degree of parallelism’ (MAXDOP)](http://blogs.msdn.com/b/chandrur/archive/2009/06/23/team-foundation-server-performance-sql-server-configuration-settings.aspx) to a value other than ‘0’ (a single command can peg all CPUs and starve other commands). The trade-off here is slower execution time vs. higher concurrency of multiple commands from multiple users.
* Consider these SQL startup traceflags. Remember, the answer to “should I do this on all my servers?” is not “yes”, the answer is “it depends on the situation”.
  + T1211 (prevent table lock escalation) ([KB934005](http://support.microsoft.com/kb/934005) and [here](http://blogs.msdn.com/b/sqlserverstorageengine/archive/2006/05/17/lock-escalation.aspx))
  + T1118 (reduce tempdb contention, [Paul says everyone should turn it on, there’s no downside.](http://www.sqlskills.com/blogs/paul/misconceptions-around-tf-1118/))
  + T1222 ([XML deadlock graph](https://www.simple-talk.com/sql/database-administration/handling-deadlocks-in-sql-server/), you’re unlikely to get deadlocks because we find most of them while dogfooding, but this information is useful if you do hit them.)
  + T1117 (equal file autogrowth for tempdb files).
* [Configure daily SQL ErrorLog rollover and 30 day retention](http://sqlblog.com/blogs/jonathan_kehayias/archive/2010/03/03/setting-sql-server-errorlog-retention-and-rollover-with-powershell.aspx).
* Set an appropriate ‘max server memory’ value for SQL server. If it’s a server dedicated to SQL (assuming TFS, SSRS and SSAS are on different machines), then [a loose formula you can use](http://www.sqlskills.com/blogs/jonathan/how-much-memory-does-my-sql-server-actually-need/) is to reserve: 1 GB of RAM for the OS, 1 GB for each 4 GB of RAM installed from 4–16 GB, and then 1 GB for every 8 GB RAM installed above 16 GB RAM. So, for a 32GB dedicated server, that’s 32-1-4-2=25GB. If you are running SSRS/SSAS/TFS on the same hardware, then you will need to reduce the amount further.