

2 Konfigurieren und Optimieren von Microsoft SQL Server

Microsoft SQL Server 2000 wurde konzipiert, um die Arbeitsauslastung dynamisch auszugleichen und um die Konfigurationseinstellungen zu optimieren. SQL Server kann beispielsweise die Arbeitsspeicherverwendung auf Grundlage der gesamten Systemspeichieranforderungen dynamisch erhöhen oder verringern.

SQL Server verwaltet außerdem den Arbeitsspeicher effizient. Dies ist besonders bei Abfragen und Benutzerverbindungen der Fall. Der Arbeitsspeicher ist nur einer von zahlreichen Bereichen, in denen die Konfiguration automatisch angepasst wird.

Die automatische Optimierung funktioniert zwar in den meisten Fällen, es gibt jedoch Situationen, in denen die SQL Server-Einstellungen manuell konfiguriert werden müssen. Wenn Sie beispielsweise eine große Datenbank mit speziellen Einschränkungen verwenden und die Datenbank nicht wie erwartet ausgeführt wird, muss die Konfiguration angepasst werden. Gegebenenfalls müssen auch die Konfigurationseinstellungen für die Konten, die Authentifizierung und die Überwachung von SQL Server geändert werden. Zum Ändern von Konfigurationseinstellungen werden hauptsächlich folgende Tools eingesetzt:

- **SQL Server Setup** Sie können unter anderem neue Instanzen von SQL Server erstellen, Komponenten hinzufügen, die SQL Server-Registrierung neu erstellen und SQL Server deinstallieren.
- **SQL Server Enterprise Manager** Enthält eine benutzerfreundliche Oberfläche, die die Datenbank- und Registrierungseinstellungen aktualisiert.
- **Gespeicherte Prozeduren** Sie können die Konfigurationseinstellungen über gespeicherte Prozeduren, beispielsweise **sp_configure** und **xp_regwrite**, anzeigen und verwalten. Beachten Sie, dass Sie einige Optionen von **sp_configure** nur dann ändern können, wenn **show advanced options** auf 1 gesetzt ist. Beispiel:

```
exec sp_configure 'show advanced options', 1
```
- **SQLSERVER.EXE** Startet SQL Server über die Befehlszeile, und Sie können die Konfigurationsparameter beim Start festlegen.

In diesem Kapitel wird ein taskorientierter Ansatz verwendet, um die Verwendung dieser Tools zum Konfigurieren von SQL Server zu erläutern.

Ausführen und Ändern von SQL Server Setup

Bei SQL Server Setup handelt es sich um ein Dienstprogramm zum Ausführen wichtiger Installationstasks für SQL Server. Sie haben mit SQL Server Setup folgende Möglichkeiten:

- Erstellen von neuen Instanzen von SQL Server
- Hinzufügen oder Entfernen von Komponenten
- Erneutes Erstellen der SQL Server-Registrierung
- Deinstallieren von SQL Server
- Erstellen einer Installationsdatei für unbeaufsichtigte SQL Server-Installationen

Erstellen von neuen Instanzen von SQL Server

Sie können mehrere Instanzen des SQL Server 2000-Datenbankmoduls auf einem Computer installieren. Das Ausführen mehrerer Instanzen des Datenbankmoduls ist optimal, wenn

- Sie mehrere Test- und Entwicklungsumgebungen auf einem großen Server unterstützen müssen.
- Sie mehrere Anwendungen auf einem Desktop ausführen müssen, wobei jede Anwendung eine eigene Instanz des SQL Server 2000-Desktopmoduls installiert.
- Sie die Datenbanken auf einem Server sicher voneinander trennen müssen.

In den meisten anderen Situationen sollten Sie jedoch nicht mehrere Instanzen des SQL Server 2000-Datenbankmoduls ausführen. Jede Instanz des SQL Server 2000-Datenbankmoduls hat eigene System- und Benutzerdatenbanken. Jede Instanz verfügt außerdem über separate SQL Server- und SQL Server-Agent-Dienste. Alle anderen Komponenten und Dienste werden gemeinsam genutzt, wodurch auf dem Server ein Overhead entsteht.

Erläuterungen zu SQL Server-Instanzen

Wenn Sie SQL Server 2000 installieren, können Sie eine Standardinstanz des SQL Server 2000-Datenbankmoduls oder eine benannte Instanz des SQL Server 2000-Datenbankmoduls installieren. In den meisten Fällen empfiehlt es sich, zuerst die Standardinstanz und anschließend nach Bedarf zusätzliche benannte Instanzen des SQL Server-Datenbankmoduls

zu installieren. Sie können eine unbegrenzte Anzahl an benannten Instanzen auf einem Computer ausführen.

Eine Standardinstanz wird über den Namen des Computers identifiziert, auf dem das SQL Server 2000-Datenbankmodul ausgeführt wird. Die Standardinstanz hat keinen separaten Instanznamen. Die Anwendungen stellen die Verbindung zur Standardinstanz über den Computernamen in ihren Anfragen her. Es kann pro Computer nur eine Standardinstanz ausgeführt werden. Bei dieser Standardinstanz kann es sich um eine beliebige Version von SQL Server handeln.

Alle anderen Instanzen von SQL Server werden durch den Instanznamen identifiziert, den Sie während der Installation festlegen. Anwendungen stellen die Verbindung zu einer benannten Instanz über den Computernamen und den Instanznamen im Format *Computername\Instanzname* her. Es kann nur das SQL Server 2000-Datenbankmodul als benannte Instanz ausgeführt werden. Frühere Versionen von SQL Server unterstützen keine benannten Instanzen.

HINWEIS *Wenn Sie SQL Server 2000 Enterprise Edition ausführen, können Sie Servercluster erstellen, die bis zu vier Server auf einem Knoten verbinden. Die Anwendungen stellen eine Verbindung zur Standardinstanz auf einem SQL Server-Cluster her, indem sie den virtuellen Servernamen angeben. Zu einer benannten Instanz auf einem SQL Server-Cluster stellen die Anwendungen die Verbindung her, indem sie den virtuellen Servernamen und die benannte Instanz im Format Virtueller_Servername\Instanzname eingeben.*

Installieren einer SQL Server-Instanz

So installieren Sie eine Instanz des SQL Server 2000-Datenbankmoduls:

1. Melden Sie sich beim Server über ein Konto mit Administratorrechten an. Legen Sie anschließend die SQL Server 2000-CD-ROM in das CD-ROM-Laufwerk ein.

TIPP *Notieren Sie sich die in den Schritten 6 bis 9 ausgeführten Aktionen. Diese Aktionen geben den Status des Servers, der Serverinstanz und der Installationsoptionen genau an. Sie benötigen diese Aufzeichnungen, falls Sie die SQL Server-Registrierungseinstellungen für die Serverinstanz neu erstellen müssen.*

2. Wenn der automatische Begrüßungsdialog aktiviert ist, startet SQL Server 2000 Setup automatisch. Doppelklicken Sie andernfalls im Stammverzeichnis der CD-ROM auf **AUTORUN.EXE**.
3. Klicken Sie auf **SQL Server 2000-Komponenten** und wählen Sie anschließend **Datenbankserver installieren**. Der Microsoft SQL Server Installations-Assistent wird aufgerufen. Klicken Sie auf **Weiter**.
4. Sie können SQL Server 2000 auf lokalen oder auf Remotecomputern installieren. Wählen Sie die Option **Lokaler Computer** oder **Remote-computer**. Klicken Sie auf **Weiter**.

HINWEIS Sie können SQL Server 2000 auch als Bestandteil eines Clusters installieren. Falls ein Cluster ermittelt wird, wird die Option **Virtueller Server** standardmäßig aktiviert.

5. Aktivieren Sie das Optionsfeld **Eine neue Instanz von SQL Server erstellen** und klicken Sie anschließend auf **Weiter**. Geben Sie anschließend die Benutzerinformationen ein und bestätigen Sie den Lizenzvertrag.
6. Aktivieren Sie das Optionsfeld **Server- und Clienttools** und klicken Sie anschließend auf **Weiter**. Wie in Abbildung 2.1 dargestellt, müssen Sie nun den zu installierenden Instanztyp bestimmen. Um eine Standardinstanz von SQL Server zu installieren, wählen Sie **Standard** und klicken anschließend auf **Weiter**. Deaktivieren Sie andernfalls das Kontrollkästchen **Standard** und geben Sie anschließend den Namen der Instanz im entsprechenden Feld ein. Der Instanzname kann bis zu 16 Zeichen lang sein und muss den Regeln für nicht begrenzte Bezeichner entsprechen.

HINWEIS Sie können auf einem Computer nur eine Standardinstanz installieren. Wenn bereits eine Standardinstanz vorhanden ist, können Sie das Kontrollkästchen **Standard** nicht aktivieren.



Abbildung 2.1: Verwenden Sie die Optionen, um den entsprechenden Instanztyp (**Standard** oder **benannt**) auszuwählen.

7. Wählen Sie im Dialogfeld **Setup-Typ** die gewünschte Installationsart aus: **Standard**, **Minimum** oder **Benutzerdefiniert**. Bei der benutzerdefinierten Installation können Sie auch Unterkomponenten und Netzwerkbibliotheken auswählen. Sie können außerdem den Standardzeichensatz und die Sortierung auswählen.

ACHTUNG Wenn Sie SQL Server mit der Installationsoption **Standard** oder **Minimum** ausführen, sind der **Standardzeichensatz** und die **Sortierung** festgelegt. Sie können diese Einstellungen bei einer

vorhandenen SQL Server-Installation nur ändern, indem Sie die master-Datenbank neu erstellen. Hierbei werden alle anderen Datenbanken vom Server getrennt und können nicht mehr verwendet werden. Weitere Informationen hierzu finden Sie im Abschnitt »Ändern der Sortierung und Neuerstellen der master-Datenbank« in diesem Kapitel.

- Legen Sie im Dialogfeld **Dienstkonto** fest, wie die SQL Server- und die SQL Server-Agent-Dienste ausgeführt werden sollen (siehe Abbildung 2.2). Um die gleichen Kontoinformationen zu verwenden und jeden Dienst automatisch zu starten, aktivieren Sie das Optionsfeld **Dasselbe Konto für jeden Dienst verwenden** und legen anschließend die Optionen im Bereich **Diensteinstellungen** fest. Aktivieren Sie andernfalls das Optionsfeld **Einstellungen für jeden Dienst anpassen**, und legen Sie anschließend die Optionen für SQL Server- und SQL Server-Agent in den Bereichen **Dienste** und **Diensteinstellungen** fest.

TIPP *SQL Server und SQL Server-Agent können im Konto Lokales System oder in einem Konto für einen Domänenbenutzer ausgeführt werden. Wenn der Server nur Ressourcen auf dem lokalen Server benötigt, verwenden Sie das Konto Lokales System. Verwenden Sie andernfalls ein Domänenkonto.*

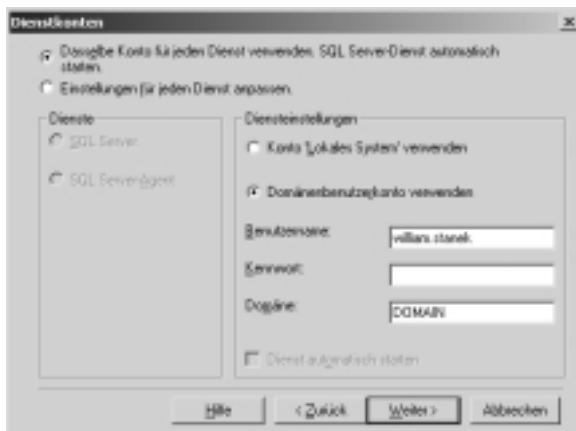


Abbildung 2.2: Legen Sie im Dialogfeld **Dienstkonto** fest, wie die SQL Server- und SQL Server-Agent-Dienste ausgeführt werden sollen.

- Konfigurieren Sie im Dialogfeld **Authentifizierungsmodus** die Authentifizierungseinstellungen. Die SQL Server-Instanz kann unter der Microsoft Windows-Authentifizierung oder unter der Authentifizierung im gemischten Modus ausgeführt werden. Bei der Windows-Authentifizierung verwenden Sie nur Domänenkonten, um die Verbindungen zur SQL Server-Instanz zu authentifizieren. Bei der Authentifizierung im gemischten Modus können Benutzer auf die SQL Server-Instanz über Domänenkonten oder SQL Server-IDs zugreifen.

10. Klicken Sie auf **Weiter**, wählen Sie den geeigneten Lizenzmodus und klicken Sie anschließend auf **Weiter**, um den Installationsprozess einzuleiten. Sobald Tasks Dateien benötigen, die von Setup installiert werden müssen, wird eine Eingabeaufforderung angezeigt. In der Eingabeaufforderung wird angegeben, welche Tasks beendet werden müssen, um einen Neustart nach der Installation zu vermeiden. Halten Sie die entsprechenden Tasks an. Wenn Sie Dienste beenden, beachten Sie, dass auch davon abhängige Dienste beendet werden. Sie müssen diese Dienste später neu starten. Klicken Sie auf **Weiter** und anschließend auf **Fertig stellen**, um den Vorgang abzuschließen.
11. Registrieren Sie die Instanz des SQL Server-Datenbankmoduls entsprechend der Anleitung in Kapitel 3, »Verwalten von Servern«. Sie können anschließend die Serverinstanz in Enterprise Manager verwalten.

Hinzufügen von Komponenten

SQL Server verfolgt, welche Komponenten installiert bzw. nicht installiert werden. Um Komponenten hinzuzufügen, gehen Sie folgendermaßen vor:

1. Melden Sie sich beim Server über ein Konto mit Administratorrechten an. Legen Sie anschließend die SQL Server 2000-CD-ROM in das CD-ROM-Laufwerk ein.
2. Wenn der automatische Begrüßungsdialog aktiviert ist, startet SQL Server 2000 Setup automatisch. Doppelklicken Sie andernfalls im Stammverzeichnis der CD-ROM auf **AUTORUN.EXE**.
3. Klicken Sie auf **SQL Server 2000-Komponenten** und wählen Sie anschließend **Datenbankserver installieren**. Der Microsoft SQL Server-Installations-Assistent wird aufgerufen. Klicken Sie auf **Weiter**.
4. Wählen Sie die Option **Lokaler Computer**, **Remotecomputer** oder **Virtueller Server**. Klicken Sie auf **Weiter**.
5. Aktivieren Sie das Optionsfeld **Eine vorhandene Instanz von SQL Server aktualisieren, entfernen oder Komponenten hinzufügen** und klicken Sie anschließend auf **Weiter**.
6. Wählen Sie im Dialogfeld **Instanzname** die gewünschte Instanz. Deaktivieren Sie gegebenenfalls das Kontrollkästchen **Standard**, damit Sie die Auswahlliste **Instanzname** verwenden können. Klicken Sie auf **Weiter**.
7. Aktivieren Sie das Optionsfeld **Komponenten zur vorhandenen Installation hinzufügen** und klicken Sie anschließend auf **Weiter**. Sie können jetzt der Installation Komponenten hinzufügen.

Das Installationsprogramm ermittelt aktuell installierte Komponenten, und Sie können anschließend zusätzliche Komponenten auswählen. Sie können unter anderem folgende Komponenten über das Installationsprogramm hinzufügen:

- **Serverkomponenten** SQL Server, Aktualisierungstools, Replikationsunterstützung, Volltextsuche, Debugsymbole und Leistungsindikatoren
- **Verwaltungstools** Enterprise Manager, Query Analyzer, Profiler, DTC-Clientunterstützung und Konflikthanzeige
- **Onlinedokumentation** Hilfedokumentation, die auf Festplatte installiert bzw. über die CD-ROM aufgerufen werden kann
- **Entwicklungstools** Header und Bibliotheken, API zum Sichern/Wiederherstellen und die Debuggerschnittstelle

Hinweis *Bereits installierte Komponenten werden nicht neu installiert. Wenn Sie in diesem Dialogfeld Komponenten deaktivieren, werden diese nicht entfernt.*

Die CD-ROM für SQL Server enthält weitere Tools, die nicht Bestandteil der Standardinstallation sind. Sie können auf diese Tools über das erste Installationsfenster zugreifen. Hierzu gehören:

- **Analysis Services** Diese Dienste werden in Verbindung mit Anwendungen für die Onlineanalyse und das Data Warehousing verwendet.
- **English Query** Dieser Dienst wird in Verbindung mit Anwendungen verwendet, die englische Abfragen der SQL-Datenbank ermöglichen.

Wiederherstellen der SQL Server-Registrierung

Die Einstellungen aller Instanzen des SQL Server 2000-Datenbankmoduls, die auf dem Computer ausgeführt werden, sind in der Windows-Registrierung gespeichert. Wenn diese Einstellungen beschädigt oder versehentlich gelöscht werden, können Sie diese über SQL Server Setup wiederherstellen. Setup erstellt die Registrierungseinstellungen auf Grundlage der von Ihnen eingegebenen Informationen neu und kopiert keine Dateien. Es ist unbedingt erforderlich, dass Sie noch wissen, wie Sie die beschädigte Instanz von SQL Server installiert haben. Wenn Ihnen dies nicht mehr bekannt ist, müssen Sie SQL Server deinstallieren und anschließend neu installieren.

HINWEIS *Die Registrierungseinstellungen gelten nur für die spezifische Instanz des SQL Server 2000-Datenbankmoduls, das Sie identifizieren. Der Wiederherstellungsprozess stellt keine Registrierungseinstellungen für andere Anwendungen und Dienste her.*

So können Sie die SQL Server-Registrierungseinstellungen neu erstellen:

1. Melden Sie sich beim Server über ein Konto mit Administratorrechten an. Legen Sie anschließend die SQL Server 2000-CD-ROM in das CD-ROM-Laufwerk ein.
2. Wenn der automatische Begrüßungsdialog aktiviert ist, startet SQL Server 2000 Setup automatisch. Doppelklicken Sie andernfalls im Stammverzeichnis der CD-ROM auf **AUTORUN.EXE**.

3. Klicken Sie auf **SQL Server 2000-Komponenten**, und wählen Sie anschließend **Datenbankserver installieren**. Der Microsoft SQL Server-Installations-Assistent wird aufgerufen. Klicken Sie auf **Weiter**.
4. Wählen Sie die Option **Lokaler Computer, Remotecomputer** oder **Virtueller Server**. Klicken Sie auf **Weiter**.
5. Wählen Sie **Erweiterte Optionen**, und klicken Sie anschließend auf **Weiter**. Aktivieren Sie das Optionsfeld **Registrierung wiederherstellen**, und klicken Sie anschließend erneut auf **Weiter**.
6. Fahren Sie mit dem Installationsprozess fort. Wählen Sie die gleichen Optionen, die Sie zuvor für die Instanz des SQL Server-Datenbankmoduls festgelegt haben. Wenn Sie die Optionen festgelegt haben, leitet Setup den Wiederherstellungsprozess ein. Wenn der Prozess erfolgreich durchgeführt wurde, wird ein Bestätigungsdialogfeld angezeigt. Andernfalls werden Sie benachrichtigt, dass Sie SQL Server für diese Datenbankinstanz deinstallieren und anschließend erneut installieren müssen.

Deinstallieren von SQL Server

Mit dem Setupprogramm von SQL Server können Sie SQL Server 2000 deinstallieren. Sie können jede Instanz des SQL Server-Datenbankmoduls separat deinstallieren.

So deinstallieren Sie eine Instanz von SQL Server:

1. Melden Sie sich beim Server über ein Konto mit Administratorrechten an. Legen Sie anschließend die SQL Server 2000-CD-ROM in das CD-ROM-Laufwerk ein.
2. Wenn der automatische Begrüßungsdialog aktiviert ist, startet SQL Server 2000 Setup automatisch. Doppelklicken Sie andernfalls im Stammverzeichnis der CD-ROM auf **AUTORUN.EXE**.
3. Klicken Sie auf **SQL Server 2000-Komponenten**, und wählen Sie anschließend **Datenbankserver installieren**. Der Microsoft SQL Server-Installations-Assistent wird aufgerufen. Klicken Sie auf **Weiter**.
4. Wählen Sie die Option **Lokaler Computer, Remotecomputer** oder **Virtueller Server**. Klicken Sie auf **Weiter**.
5. Aktivieren Sie das Optionsfeld **Eine vorhandene Instanz von SQL Server aktualisieren, entfernen oder Komponenten hinzufügen**, und klicken Sie anschließend auf **Weiter**.
6. Wählen Sie im Dialogfeld **Instanzname** die gewünschte Instanz, und klicken Sie anschließend auf **Weiter**.
7. Aktivieren Sie das Optionsfeld **Die vorhandene Installation deinstallieren**, und klicken Sie anschließend auf **Weiter**. Der Installations-Assistent entfernt die SQL Server-Instanz.
8. Klicken Sie auf **Weiter** und anschließend auf **Fertig stellen**, um den Vorgang abzuschließen.

Konfigurieren von SQL Server mit Enterprise Manager

Enterprise Manager bietet die einfachste Möglichkeit zum Konfigurieren von SQL Server. In Enterprise Manager rufen Sie das Dialogfeld **Eigenschaften** eines registrierten Servers auf und konfigurieren anschließend die entsprechenden Registerkarten und Optionen für den Server. SQL Server führt im Hintergrund Befehle aus, die die Servereinstellungen ändern. Wenn diese Änderungen die **master**-Datenbank oder andere Datenbanken betreffen, wird die gespeicherte Prozedur **sp_configure** verwendet, um in der Tabelle **sysconfigures** Änderungen vorzunehmen. Wenn die Änderungen die Microsoft Windows-Registrierung betreffen, aktualisiert die gespeicherte Prozedur **xp_regwrite** die Registrierung. SQL Server verwendet gespeicherte Prozeduren auch zum Lesen der Konfigurationseinstellungen. Die gespeicherte Prozedur **xp_regread** wird beispielsweise zum Lesen der meisten Registrierungseinstellungen verwendet.

Nachdem Sie einen Server in Enterprise Manager registriert haben, können Sie eine Verbindung herstellen und deren Konfiguration im Dialogfeld **SQL Server-Eigenschaften** verwalten. So greifen Sie auf dieses Dialogfeld zu:

1. Wählen Sie **Start**, danach **Microsoft SQL Server** und anschließend **Enterprise Manager**.
2. Rufen Sie die Servergruppe auf, die den gewünschten Server enthält. Klicken Sie in der Strukturansicht von Enterprise Manager mit der rechten Maustaste auf den Servernamen und klicken Sie im Kontextmenü auf **Eigenschaften**. Das in Abbildung 2.3 dargestellte Dialogfeld wird angezeigt.
3. Sie können jetzt allgemeine SQL Server-Einstellungen konfigurieren. Um erweiterte Einstellungen festlegen zu können, müssen Sie eine gespeicherte Prozedur verwenden, beispielsweise **sp_configure**.

Das Dialogfeld **SQL Server-Eigenschaften** enthält viele Registerkarten. In den folgenden Abschnitten werden diese Registerkarten erläutert. Um eine Zusammenfassung der aktuellen Einstellungen in SQL Server Query Analyzer anzuzeigen, führen Sie folgenden Befehl aus:

```
exec sp_configure
```



Abbildung 2.3: Sie können im Dialogfeld **SQL Server-Eigenschaften** SQL Server ohne eine gespeicherte Prozedur wie **sp_configure** ausführen. Dieses Dialogfeld kann mit der Maus bedient werden und ist somit sehr benutzerfreundlich.

Bestimmen von System- und Serverinformationen

Auf der Registerkarte **Allgemein** des Dialogfeldes **SQL Server-Eigenschaften** finden Sie allgemeine System- und Serverinformationen (siehe Abbildung 2.3). Mithilfe der auf der Registerkarte **Allgemein** verfügbaren Informationen können Sie Folgendes bestimmen:

- Die Edition von SQL Server: Standard, Enterprise oder Personal
- Version des Betriebssystems
- SQL Server-Version
- Standardsprache
- Plattform- und Chiparchitektur
- Größe des auf dem System installierten Arbeitsspeichers
- Anzahl der CPUs
- Speicherort des SQL Server-Stammverzeichnisses
- Standardserversortierung

Sie können über die erweiterte gespeicherte Prozedur **xp_msver** vergleichbare Informationen aufrufen. Führen Sie dazu den folgenden Befehl aus:

```
exec xp_msver 'ProductName', 'ProductVersion', 'Language', 'Platform',
'WindowsVersion', 'PhysicalMemory', 'ProcessorCount'
```

TIPP Sie können mit dem **SQL Query Analyzer** den angezeigten Befehl ausführen. Weitere Informationen zu Basistechniken für dieses Dienstprogramm finden Sie im Abschnitt »Konfigurieren von SQL Server mit gespeicherten Prozeduren« in diesem Kapitel.

Konfigurieren des Starts

SQL Server bietet zahlreiche Möglichkeiten zum Konfigurieren des Serverstarts. Eine davon ist die Registerkarte **Allgemein** im Dialogfeld **SQL Server-Eigenschaften**. Verwenden Sie die Kontrollkästchen im Bereich **Autostarttrichtlinien für Betriebssystemstart**. Über diese Kontrollkästchen bestimmen Sie, ob die zugehörigen Dienste automatisch beim Start des Betriebssystems ausgeführt werden sollen. Sie können über diese Kontrollkästchen Folgendes steuern:

- **Konfigurieren des SQL Server-Autostarts** Um SQL Server (MSSQL-Server-Dienst) beim Systemstart automatisch zu starten, aktivieren Sie das Kontrollkästchen **SQL Server automatisch starten**. Deaktivieren Sie andernfalls dieses Kontrollkästchen.
- **Konfigurieren des SQL Server-Agent-Autostarts** Um SQL Server-Agent (SQLServerAgent-Dienst) beim Systemstart automatisch zu starten, aktivieren Sie das Kontrollkästchen **SQL Server-Agent automatisch starten**. Deaktivieren Sie andernfalls dieses Kontrollkästchen.

HINWEIS Die **MSSQLServer-** und **SQLServerAgent-Dienste** werden nur mit der Standardserverinstanz verwendet. Wenn Sie mit einer benannten Instanz von SQL Server arbeiten, werden diese Dienste in **MSSQL\$Instanzname** und **SQLAgent\$Instanzname** umbenannt.

- **Konfigurieren des MSDTC-Autostarts** Um MSDTC (MSDTC-Dienst) beim Systemstart automatisch zu starten, aktivieren Sie das Kontrollkästchen **MSDTC automatisch starten**. Deaktivieren Sie andernfalls dieses Kontrollkästchen.

Enterprise Manager verwendet die gespeicherte Prozedur **xp_regwrite**, um die Autostarteinstellungen für Dienste zu ändern. Diese Einstellungen betreffen die Autostartwerte der Registrierungsschlüssel.

ACHTUNG Schreiben Sie nur dann direkt in die Registrierung, wenn dies absolut erforderlich ist. Verwenden Sie hierzu nach Möglichkeit immer das Dialogfeld **Eigenschaften**.

Einrichten von Startparametern

Startparameter steuern das Startverhalten von SQL Server und die dabei festgelegten Optionen. Die einfachste Möglichkeit zum Konfigurieren von Startoptionen ist die Registerkarte **Allgemein** im Dialogfeld **SQL Server-Eigenschaften**. Klicken Sie auf **Startparameter**, um ein Dialogfeld anzuzeigen (siehe Abbildung 2.4). Sie können jetzt Startparameter hinzufügen und entfernen.

TIPP Startparameter können auch an das befehlszeilenorientierte Dienstprogramm SQLSERVER.EXE übergeben werden. Wenn diesem Dienstprogramm die Option **-c** übergeben wird, startet SQL Server ohne einen Dienst. Sie müssen SQLSERVER.EXE im Verzeichnis Binn ausführen, das der Instanz des SQL Server-Datenbankmoduls entspricht, das Sie starten möchten. Bei der Standardinstanz befindet sich der Dienst im Verzeichnis mssql\Binn. Bei benannten Instanzen befindet sich das Dienstprogramm im Verzeichnis \mssql\$Instanzname\Binn.

Hinzufügen von Startparametern

So können Sie Startoptionen hinzufügen:

1. Geben Sie im Feld **Parameter** den gewünschten Parameter ein.
2. Klicken Sie auf **Hinzufügen**.
3. Klicken Sie auf **OK**, um die Änderungen zu übernehmen.

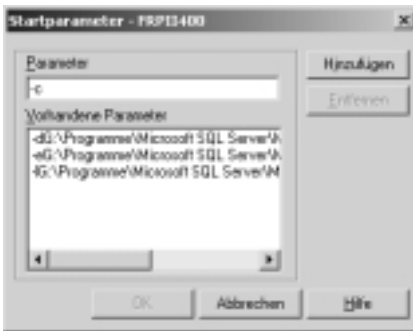


Abbildung 2.4: Konfigurieren Sie die Startparameter für SQL Server, indem Sie Optionen hinzufügen oder entfernen. Löschen Sie die Standardoptionen (-d, -e und -l) nicht, weil andernfalls Probleme auftreten können.

Entfernen von Startparametern

So entfernen Sie Startparameter:

1. Wählen Sie den gewünschten Parameter, und klicken Sie anschließend auf **Entfernen**.
2. Klicken Sie auf **OK**, um die Änderungen zu übernehmen.

Allgemeine Startparameter

In Tabelle 2.1 werden die Startparameter erläutert. Die ersten drei Parameter (-d, -e und -l) sind die Standardparameter für SQL Server. Sie können über die verbleibenden Parameter zusätzliche Einstellungen festlegen.

Tabelle 2.1: Startparameter für SQL Server

Parameter	Beschreibung	Beispiel
-d<Pfad>	Legt den vollständigen Pfad für die master -Datenbank fest.	-dC:\MSSQL\data\master.mdf
-e<Pfad>	Legt den vollständigen Pfad für das Fehlerprotokoll fest.	-eD:\LOGS\ERRORLOG
-l<Pfad>	Legt den vollständigen Pfad für das Transaktionsprotokoll der master -Datenbank fest.	-lD:\LOGS\mastlog.ldf
-B	Setzt einen Haltepunkt bei einem Fehler. Wird mit der Option -y beim Debugging verwendet.	
-c	SQL Server wird nicht als Dienst ausgeführt. Dadurch wird der Start beschleunigt, wenn Sie SQL Server über die Befehlszeile aufrufen.	
-f	Startet SQL Server mit der Minimal-konfiguration. Aktiviert die Option sp_configure (allow updates) , die standardmäßig deaktiviert ist.	
-g Anzahl	Gibt den Arbeitsspeicher (in MB) an, der für SQL Server reserviert werden soll.	-g 192
-m	Startet SQL Server im Einzelbenutzermodus. In diesem Modus kann nur ein Benutzer eine Verbindung herstellen. Außerdem wird der Prüfpunktprozess nicht gestartet.	
-n	Weist SQL Server an, keine Fehler in das Ereignisprotokoll der Anwendung zu schreiben.	
-p<Ebene>	Legt die Genauigkeit für numerische und dezimale Datentypen fest. Der Standard ist 38 Stellen. Der zulässige Bereich ist 1 bis 38. Es wird die maximale Genauigkeit (38) verwendet, wenn Sie die Option ohne Festlegen einer Genauigkeit aktivieren.	-p38
-s<Alternativer Registrierungs-schlüssel>	Legt einen alternativen Registrierungs-schlüssel für SQL Server fest. Dieser Schlüssel wird während des Starts anstelle des Standardschlüssels verwendet.	-sHKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSSQLServerMod
-T<Wert für Ablaufverfolgung>	Setzt ein Flag für die Ablaufverfolgung. Flags für die Ablaufverfolgung legen ein vom Standard abweichendes Verhalten fest und werden oft beim Debugging oder Diagnostizieren von Leistungsproblemen verwendet.	-T237



Parameter	Beschreibung	Beispiel
-t<Wert für Ablaufverfolgung>	Setzt ein internes Flag für die Ablaufverfolgung für SQL Server. Wird nur vom technischen Support für SQL Server verwendet.	-t8837
-x	Deaktiviert die Statistikprotokollierung für die CPU-Zeit und die Cachetrefferquote. Ermöglicht eine maximale Leistung.	
-y Nummer	Setzt eine Fehlernummer, die dafür sorgt, dass SQL Server den Stack sichert.	-y 1803

Einrichten des Dienststartkontos

SQL Server erbt Rechte und Berechtigungen vom Dienststartkonto. Diese Berechtigungen werden immer dann verwendet, wenn SQL Server Tasks auf dem lokalen System oder über das Netzwerk ausführt. Sie können zwei Kontenarten verwenden (Weitere Informationen hierzu finden Sie im Abschnitt »Dienstkonten« in Kapitel 1): Lokale Systemkonten und Domänenkonten. Wenn SQL Server nur lokale Vorgänge ausführt, verwenden Sie das Konto **Lokales System**. Verwenden Sie andernfalls ein korrekt konfiguriertes Domänenkonto.

Hinweis *Wenn Sie SQL Mail oder SQLAgent mail verwenden möchten, stellen Sie nach Möglichkeit sicher, dass das Domänenbenutzerkonto für die Ausführung von SQL Server und SQL Server-Agent ein Mitglied der lokalen Administratorgruppe ist und dass beiden Diensten das gleiche Konto zugeordnet wurde. Wenn Sie das Konto nach der Installation von SQL Server ändern, sind verschiedene Änderungen der Sicherheitseinstellungen erforderlich, um die Dienste neu starten zu können.*

Verwenden des Kontos »Lokales System«

So können Sie das Konto **Lokales System** für den SQL Server-Start angeben:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Sicherheit**.
2. Verwenden Sie das Konto **Lokales System**. Aktivieren Sie hierzu im Bereich **Dienststartkonto** das Optionsfeld **Systemkonto**.
3. Klicken Sie auf **OK**. Jetzt kann der Server nur auf lokale Ressourcen zugreifen.

Verwenden eines Domänenkontos

So können Sie ein Domänenkonto für den SQL Server-Start angeben:

1. Legen Sie die geeigneten Berechtigungen für das SQL Server-Konto fest, und vergewissern Sie sich, dass für das Benutzerkonto die lokale Sicherheitsrichtlinie **Anmelden als Dienst** aktiviert ist.
2. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Sicherheit**.
3. Aktivieren Sie das Optionsfeld **Dieses Konto**. Geben Sie anschließend den entsprechenden Kontonamen und das Kennwort ein. Sie können auch die Domäne als Bestandteil des Kontonamens angeben, beispielsweise **cordata\wrstane**. In diesem Fall ist **cordata** der Domänenname und **wrstane** der Kontoname.
4. Klicken Sie auf **OK**.

Authentifizierung und Überwachung

Sie konfigurieren die Authentifizierungs- und Überwachungsoptionen im Dialogfeld **SQL Server-Eigenschaften** auf der Registerkarte **Sicherheit**. Diese Registerkarte wird in Abbildung 2.5 dargestellt.

Einrichten des Authentifizierungsmodus

Wie bereits im Abschnitt »Authentifizierungserweiterungen« von Kapitel 1 erläutert, kann SQL Server eine kombinierte Domänen- und SQL Server-Authentifizierung oder nur eine Domänenauthentifizierung verwenden. Um die kombinierte Authentifizierung zu verwenden, aktivieren Sie das Optionsfeld **SQL Server und Windows**. Jetzt können sich Benutzer in Windows-Domänen beim Server über ein Domänenkonto und andere Benutzer über einen SQL Server-Benutzernamen anmelden.

Um nur die Domänenauthentifizierung zu verwenden, aktivieren Sie das Optionsfeld **Nur Windows**. Jetzt können sich nur Benutzer mit einem Domänenkonto beim Server anmelden.

TIPP *SQL Server prüft bei der kombinierten Authentifizierung, ob ein neuer Benutzername ein SQL Server-Benutzername ist. Wenn der Benutzername vorhanden ist, verwendet SQL Server anschließend das angegebene Kennwort, um den Benutzer zu authentifizieren. Wenn der Benutzername nicht vorhanden ist, wird die Domänenauthentifizierung verwendet. Beachten Sie außerdem, dass die Domänenauthentifizierung bei SQL Server Desktop unter Windows 95 oder Windows 98 nicht verfügbar ist.*



Abbildung 2.5: Sie können die Authentifizierung und Überwachung mit den Optionen auf der Registerkarte **Sicherheit** konfigurieren.

Einrichten der Überwachungsebene

Mithilfe der Überwachung können Sie den Benutzerzugriff auf SQL Server verfolgen. Sie können die Überwachung bei beiden Authentifizierungsmodi sowie bei vertrauenswürdigen und nicht vertrauenswürdigen Verbindungen verwenden.

Wenn die Überwachung aktiviert ist, werden Benutzeranmeldungen im Windows-Anwendungsprotokoll und/oder im SQL Server-Fehlerprotokoll erfasst (abhängig davon, wie Sie die Protokollierung für SQL Server konfiguriert haben). Folgende Überwachungsoptionen stehen zur Verfügung:

- **Keine** Deaktiviert die Überwachung (Standardeinstellung)
- **Erfolg** Überwacht erfolgreiche Anmeldungen
- **Fehler** Überwacht fehlgeschlagene Anmeldungen
- **Alle** Überwacht erfolgreiche und fehlgeschlagene Anmeldungen

Optimieren der Arbeitsspeicherverwendung

SQL Server verwaltet den Arbeitsspeicherbedarf dynamisch. In der Regel wird dieser Vorgang optimal ausgeführt. Bei der dynamischen Arbeitsspeicherreservierung kann SQL Server Arbeitsspeicher für eingehende Abfragen hinzufügen, Arbeitsspeicher für eine soeben gestartete Anwendung zuteilen oder Arbeitsspeicher für einen möglichen Bedarf reservieren. Es gelten folgende Standardeinstellungen für den Arbeitsspeicher:

- SQL Server-Arbeitsspeicher dynamisch konfigurieren
- Mindestreservierung von 0 MB für den Arbeitsspeicher
- Die maximale Arbeitsspeicherreservierung wird auf das RAM-Gesamtvolumen auf dem System gesetzt
- Es wird kein Arbeitsspeicher speziell für SQL Server reserviert
- Mindestarbeitsspeicher für Abfrageausführung von 1024 KB

Sie können diese Einstellungen gegebenenfalls ändern, sollten dabei aber sehr vorsichtig sein, damit für SQL Server nicht zu wenig oder zu viel Arbeitsspeicher reserviert wird. Wenn SQL Server zu wenig Arbeitsspeicher zur Verfügung steht, werden die Tasks in der Regel nicht rechtzeitig ausgeführt. Wenn Sie zu viel Arbeitsspeicher vergeben, belegt SQL Server wichtige Ressourcen, die von anderen Anwendungen benötigt werden (beispielsweise vom Betriebssystem). Dies kann zu umfangreichen Auslagerungen führen und die Gesamtleistung des Systems erheblich beeinträchtigen.

TIPP Bestimmte Statistiken dienen als Entscheidungshilfe für die Arbeitsspeicherreservierung. Hierzu gehören die Anzahl der Seitenfehler pro Sekunde sowie die Cachetrefferquote. Über Seitenfehler pro Sekunde können Sie die Auslagerungen in den bzw. aus dem virtuellen Arbeitsspeicher verfolgen. Über die Cachetrefferquote können Sie überwachen, ob sich die abgerufenen Daten im Arbeitsspeicher befinden. Weitere Informationen hierzu finden Sie in Kapitel 10, »Profilerstellung und Überwachung für Microsoft SQL Server 2000«.

Die folgenden Abschnitte untersuchen die Hauptbereiche der Speicherverwaltung. In der Regel wird die Arbeitsspeicherverwendung im Dialogfeld **SQL Server-Eigenschaften** auf der Registerkarte **Arbeitsspeicher** konfiguriert (siehe Abbildung 2.6). Es wird noch eine bessere Möglichkeit für die Konfiguration der Windows-Arbeitsspeicherverwendung für SQL Server erläutert werden.

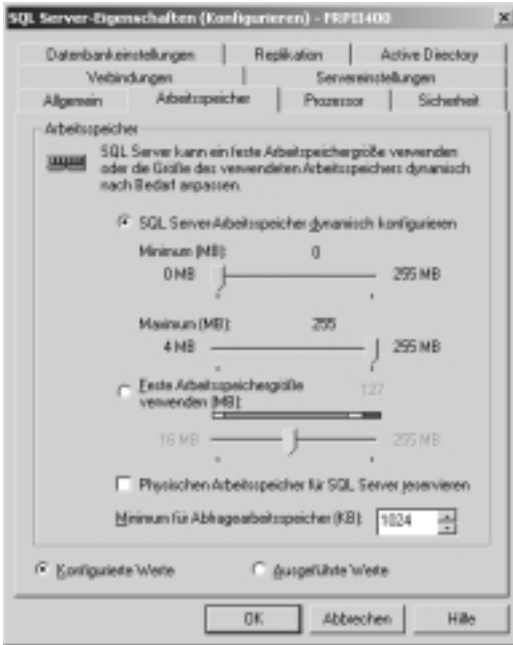


Abbildung 2.6: Sie können auf der Registerkarte **Arbeitsspeicher** die Arbeitsspeicherverwendung dynamisch oder manuell konfigurieren. Beachten Sie jedoch, dass diese Einstellungen nur bei Leistungsproblemen geändert werden sollten.

Arbeiten mit dynamisch konfiguriertem Arbeitsspeicher

SQL Server konfiguriert die Arbeitsspeicherverwendung mit dem dynamisch konfigurierten Arbeitsspeicher automatisch auf Grundlage der Arbeitsauslastung und der verfügbaren Ressourcen. Die gesamte Arbeitsspeicherverwendung liegt im Bereich der von Ihnen festgelegten Unter- und Obergrenze. So verwenden Sie den dynamisch konfigurierten Arbeitsspeicher:

1. Aktivieren Sie auf der Registerkarte **Arbeitsspeicher** das Optionsfeld **SQL Server-Arbeitsspeicher dynamisch konfigurieren**.
2. Sie können gegebenenfalls mit dem Schieberegler die Grenzen für die Arbeitsspeicherverwendung festlegen. Es wird empfohlen, dass Sie für eigenständige Werte einen Höchstwert festlegen, der dem gesamten RAM entspricht.
3. Klicken Sie auf **OK**.

Sie können über die gespeicherte Prozedur **sp_configure** die Grenzwerte für die Arbeitsspeicherverwendung ändern. Verwenden Sie folgende Transact-SQL-Befehle:

```
exec sp_configure 'min server memory', <Anzahl MB>
exec sp_configure 'max server memory', <Anzahl MB>
```

PRAKTISCHE HINWEISE Mit *dynamisch konfiguriertem Speicher* brauchen Sie in der Regel keine Ober- und Untergrenze für die Arbeitsspeicherverwendung festzulegen. Bei einem dedizierten System,

*auf dem nur SQL Server ausgeführt wird, können Sie jedoch einen besseren Betrieb gewährleisten, indem Sie einen Mindestwert von 4 MB + (24 KB * NumUsers) festlegen. Hierbei gilt: NumUsers ist die durchschnittliche Anzahl der Benutzer, die gleichzeitig mit dem Server verbunden sind. Sie können auch einen physischen Arbeitsspeicher für SQL Server reservieren. SQL Server verwendet ungefähr 3,5 MB für den Code und für interne Strukturen. Zusätzlicher Arbeitsspeicher wird folgendermaßen verwendet: 96 Byte für Sperren, 2880 Byte für offene Datenbanken und 276 Byte für offene Objekte, die alle Tabellen, Sichten, gespeicherten Prozeduren, erweiterten gespeicherten Prozeduren, Trigger, Regeln, Einschränkungen und Standards enthalten. Für die Protokollierung von zusätzlichem Arbeitsspeicher werden 500 KB benötigt. Daraus ergibt sich ein Gesamtvolumen von 4 MB.*

Verwenden von festem Arbeitsspeicher

Wenn Sie die dynamische Speicherverwaltung außer Kraft setzen möchten, können Sie für SQL Server eine feste Speichergröße festlegen. Führen Sie die folgenden Schritte aus:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Arbeitsspeicher** und aktivieren Sie das Optionsfeld **Feste Arbeitsspeichergröße verwenden**.
2. Legen Sie den gewünschten Arbeitsspeicher mit dem Schieberegler fest.
3. Klicken Sie auf **OK**.

ACHTUNG *Wenn Sie den festen Arbeitsspeicher falsch wählen, kann dadurch die Ausführung von SQL Server erheblich beeinträchtigt werden. Verwenden Sie den festen Arbeitsspeicher nur, wenn Sie sicherstellen müssen, dass für SQL Server eine genaue Arbeitsspeichergröße zur Verfügung steht.*

Reservieren von physischem Arbeitsspeicher

Sie können speziell für SQL Server Arbeitsspeicher reservieren. Wenn Sie physischen Arbeitsspeicher für SQL Server reservieren, lagert das Betriebssystem auch dann keine SQL Server-Arbeitsspeicherseiten aus, wenn der Arbeitsspeicher bei einem Leerlauf von SQL Server anderen Prozessen zugewiesen werden kann. Auf einem dedizierten System kann die Reservierung von Arbeitsspeicher die SQL Server-Leistung verbessern, weil ein geringerer Aufwand für die Auslagerung und die Cachetreffer entsteht.

So reservieren Sie physischen Arbeitsspeicher für SQL Server:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Arbeitsspeicher** und aktivieren Sie das Kontrollkästchen **Physischen Arbeitsspeicher für SQL Server reservieren**.
2. Klicken Sie auf **OK**.

Sie können physischen Arbeitsspeicher auch über die gespeicherte Prozedur **sp_configure** reservieren. Verwenden Sie hierzu folgenden Transact-SQL-Befehl:

```
exec sp_configure 'set working set size', 1
```

Reservieren von Arbeitsspeicher für Abfragen

Standardmäßig reserviert SQL Server mindestens 1024 KB Arbeitsspeicher für die Abfrageausführung. Diese Arbeitsspeicherreservierung wird pro Benutzer garantiert, und Sie können einen beliebigen Wert im Bereich von 512 KB bis 2 GB festlegen. Wenn Sie die Mindestgröße für Abfragen erhöhen, können Sie die Leistung von Abfragen verbessern, die prozessorintensive Vorgänge ausführen, beispielsweise Sortieren oder Hashing. Wenn Sie einen zu hohen Wert festlegen, kann jedoch die Gesamtleistung des Systems beeinträchtigt werden. Passen Sie daher die Mindestgröße für Abfragen nur an, wenn Abfragen zu langsam ausgeführt werden.

PRAKTISCHE HINWEISE *Die Standardeinstellung von 1024 KB RAM ist in den meisten Fällen optimal. Sie können diesen Wert jedoch ändern, wenn der Server in einer extrem beanspruchten Umgebung mit zahlreichen gleichzeitig ausgeführten Abfragen mit separaten Benutzerverbindungen betrieben wird. Eine Änderung dieses Werts ist gegebenenfalls auch in einer relativ langsamen Umgebung mit wenigen (aber großen oder komplexen) Abfragen erforderlich. In diesem Fall hängt die Entscheidung für die Anpassung der Mindestgröße für Abfragen von vier Faktoren ab: vom Gesamtvolumen des freien Arbeitsspeichers (wenn das System im Leerlauf ist und SQL Server ausgeführt wird), von der durchschnittlichen Anzahl der gleichzeitig über separate Benutzerverbindungen ausgeführten Abfragen, der durchschnittlichen Abfragegröße und von der gewünschten Antwortzeit für Abfragen. Häufig muss ein Kompromiss gefunden werden. Sie können nicht immer eine sofortige Antwort erhalten, aber Sie haben die Möglichkeit, die Leistung auf Grundlage der verfügbaren Ressourcen zu optimieren.*

Verwenden Sie die folgende Formel als Ausgangspunkt für die Optimierung:

Freier Arbeitsspeicher / (Durchschnittliche Abfragegröße * Durchschnittliche Anzahl der simultan ausgeführten Abfragen).

*Wenn dem System beispielsweise 200 MB freier Arbeitsspeicher zur Verfügung steht, die durchschnittliche Abfragegröße 2 MB beträgt und durchschnittlich 5 gleichzeitige Abfragen ausgeführt werden, beträgt der optimale Wert für die Abfragegröße 200 MB / (2*5) oder 20 MB. Allgemein stellt dieser Wert das Maximum dar, das Sie in der aktuellen Umgebung zuweisen sollten, wobei in der Regel ein geringerer Wert empfohlen wird.*

So weisen Sie Arbeitsspeicher für Abfragen zu:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Arbeitsspeicher** und legen Sie den gewünschten Wert im Feld **Minimum für Abfragearbeitsspeicher** fest. Dieser Wert wird in KB festgelegt.
2. Klicken Sie auf **OK**.

Sie können das Minimum für den Abfragearbeitsspeicher auch über die gespeicherte Prozedur **sp_configure** festlegen. Verwenden Sie hierzu folgenden Befehl:

```
exec sp_configure 'min memory per query', <Anzahl KB>
```

Konfigurieren der Prozessoren und der parallelen Verarbeitung

Multiprozessorsysteme können die verbesserten SQL Server-Features für paralleles und symmetrisches Multiprocessing nutzen. Sie können die Methode und den Zeitpunkt für die SQL Server-Verwendung der Prozessoren sowie die parallele Verarbeitung der Abfragen bestimmen. Unter Windows 2000 unterstützen die Standardversionen von SQL Server bis zu vier Prozessoren. Die Enterprise Editions unterstützen bis zu 32 Prozessoren (8 mit Advanced Server und 32 mit Data Center Server).

Optimieren der CPU-Nutzung

Die Prozesseureinstellungen werden bei der Installation von SQL Server automatisch konfiguriert und optimiert. Diese Einstellungen sollten nach Möglichkeit nur nach einer genauen Analyse geändert werden. So können Sie die Prozessorverwendung manuell konfigurieren:

1. Wählen Sie **Start**, danach **Microsoft SQL Server** und anschließend **Enterprise Manager**.
2. Klicken Sie in der Strukturansicht von Enterprise Manager mit der rechten Maustaste auf den Servernamen und klicken Sie im Kontextmenü auf **Eigenschaften**.
3. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die in Abbildung 2.7 dargestellte Registerkarte **Prozessor**.
4. In der Prozessorliste finden Sie die von SQL Server verwendeten Prozessoren. Aktivieren Sie die Kontrollkästchen für die gewünschten Prozessoren. Deaktivieren Sie die Kontrollkästchen für die Prozessoren, die nicht verwendet werden sollen. Die erste CPU des Systems heißt CPU 0, die zweite CPU 1 usw.



Abbildung 2.7: Auf der Registerkarte **Prozessor** können Sie festlegen, welche Prozessoren von SQL Server verwendet werden und wann die parallele Verarbeitung in Kraft tritt. Lesen Sie die Hinweise, bevor Sie Änderungen vornehmen.

PRAKTISCHE HINWEISE Wenn das System über mehr als die von SQL Server unterstützte Anzahl an Prozessoren verfügt, verwendet SQL Server nicht alle Prozessoren. Unter einem achtfachen SMP-System (Symmetric Multiprocessing) kann SQL Server Standard nur vier Prozessoren nutzen. Dies bedeutet, dass die restlichen vier Prozessoren anderen Anwendungen und Tasks auf Systemebene zur Verfügung stehen.

Es wird nicht empfohlen, SQL Server die Prozessoren mit den höheren Nummern (5, 6, 7 und 8) zuzuweisen. Windows ordnet den Prozessoren mit den höchsten Nummern verzögerte Prozessaufrufe mit Netzwerkschnittstellenkarten (NICs) zu. Wenn in diesem Beispiel das System zwei Netzwerkschnittstellenkarten verwendet, werden diese Aufrufe an die CPU 8 und CPU 7 weitergeleitet. Halten Sie Rücksprache mit dem Hersteller Ihres Systems, bevor Sie diese Werte ändern.

5. Klicken Sie auf **OK**.
6. Diese Einstellungen werden nach dem Neustart des Servers übernommen.

Sie können die gespeicherte Prozedur **sp_configure** verwenden, um die Affinitätsmaske zu ändern. Verwenden Sie hierzu folgenden Befehl:

```
exec sp_configure 'affinity mask', <Ganzzahl>
```

SQL Server interpretiert die Ganzzahl als Bitmaske, die die zu verwendenden Prozessoren darstellt. In dieser Bitmaske wird CPU 0 durch Bit 0,

CPU 1 durch Bit 1 usw. dargestellt. Mit dem Bitwert 1 weisen Sie SQL Server an, die CPU zu verwenden. Mit dem Bitwert 0 weisen Sie SQL Server an, die CPU nicht zu verwenden. Wenn Sie beispielsweise die Unterstützung für die Prozessoren 1, 2 und 5 aktivieren möchten, erhalten Sie den Binärwert

000100110

Die entsprechende Ganzzahl beträgt 76.

$64 + 8 + 4 = 76$

Festlegen der parallelen Verarbeitung

Um zu bestimmen, wann die parallele Verarbeitung verwendet werden soll, sind einige Berechnungen erforderlich. Allgemein werden Abfragen von SQL Server parallel ausgeführt,

- wenn die Anzahl der CPUs größer als die Anzahl der aktiven Verbindungen ist.
- wenn die geschätzten Kosten für die serielle Ausführung einer Abfrage höher sind als die Schwelle des Abfrageplanes. (Die geschätzten Kosten beziehen sich auf die verstrichene Zeit (in Sekunden), die für die serielle Ausführung der Abfrage erforderlich ist.)

Bestimmte Anweisungstypen können nur dann parallel verarbeitet werden, wenn sie Klauseln enthalten. UPDATE, INSERT und DELETE werden in der Regel selbst dann nicht parallel verarbeitet, wenn die zugehörige Abfrage den Kriterien entspricht. Wenn die UPDATE- oder DELETE-Anweisungen jedoch eine WHERE-Klausel enthalten oder eine INSERT-Anweisung eine SELECT-Klausel enthält, können WHERE und SELECT parallel ausgeführt werden. Die Änderungen werden anschließend der Datenbank seriell zugewiesen.

So konfigurieren Sie die parallele Verarbeitung:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Prozessor**.
2. Um alle verfügbaren Prozessoren für die parallele Verarbeitung (die maximal von SQL Server unterstützt werden) zu verwenden, aktivieren Sie das Optionsfeld **Alle verfügbaren Prozessoren verwenden**.
3. Um eine bestimmte Anzahl an Prozessoren für die parallele Verarbeitung (bis zu der maximal von SQL Server unterstützten Anzahl) festzulegen, aktivieren Sie das Optionsfeld **Verwenden von x Prozessor(en)** und wählen anschließend die gewünschte Anzahl der Prozessoren. Mit dem Wert 1 weisen Sie SQL Server an, keine parallele Verarbeitung durchzuführen.
4. Geben Sie im Feld **Minimum für Abfragenanzahl** einen Kostenvorschlag ein. Sie können einen beliebigen Wert im Bereich von 0 bis 32 767 eingeben. Wenn Sie nur eine CPU verwenden, wird diese Angabe ignoriert.

5. Klicken Sie auf **OK**. Diese Änderungen werden sofort übernommen. Sie müssen den Server nicht neu starten.

Sie können die parallele Verarbeitung auch über die gespeicherte Prozedur **sp_configure** konfigurieren. Die hierzu erforderlichen Transact-SQL-Befehle lauten:

```
exec sp_configure 'max degree of parallelism', <Ganzzahl> exec sp_configure  
'cost threshold for parallelism', <Ganzzahl>
```

HINWEIS *Transact-SQL ist eine erweiterte Version von SQL (Standard Structured Query Language), die SQL Server verwendet. Transact-SQL wird auch T-SQL genannt.*

Konfigurieren von Threading, Prioritäten und Fibers

Threads sind ein sehr wichtiger Bestandteil eines Multitasking-Betriebssystems und ermöglichen, dass SQL Server viele Aktionen sofort ausführen kann. Bei Threads handelt es sich jedoch nicht um Prozesse. Es handelt sich vielmehr um parallele Ausführungspfade, über die Anwendungen die CPU besser nutzen können.

SQL Server versucht, Threads auf Benutzerverbindungen abzustimmen. Wenn die Anzahl der verfügbaren Threads größer als die Anzahl der Benutzerverbindungen ist, besteht mindestens ein 1:1-Verhältnis der Threads zu Benutzerverbindungen, weshalb jede Benutzerverbindung eindeutig abgewickelt werden kann. Wenn die Anzahl der verfügbaren Threads kleiner als die Anzahl der Benutzerverbindungen ist, muss SQL Server einen Threadpool bilden. Daraus ergibt sich, dass der gleiche Thread unter Umständen mehrere Benutzerverbindungen unterstützt. Dies kann die Leistung und die Antwortzeit beeinträchtigen, wenn zusätzliche Ressourcen zwar verfügbar sind, jedoch nicht verwendet werden.

In der Regel wickelt das Betriebssystem die Threads im Kernelmodus, Anwendungen und benutzerbezogene Tasks jedoch im Benutzermodus ab. Der Wechsel zwischen den Betriebsmodi benötigt CPU-Zyklen und Ressourcen. Ein Wechsel ist beispielsweise erforderlich, wenn der Kernel einen neuen Thread abwickeln muss. Um festzulegen, dass die Anwendung das Threading direkt ausführen kann, können Sie Fibers verwenden. Wenn Sie die Fibers wechseln, brauchen Sie keine Betriebsmodi zu ändern und können somit die Leistung gelegentlich verbessern. Sie können hierzu auch die Priorität von SQL Server-Threads erhöhen. Threads haben in der Regel eine Priorität von 1 bis 31, wobei Threads mit einer höheren Priorität mehr CPU-Zeit erhalten als Threads mit einer niedrigeren Priorität. Threads mit einer höheren Priorität können außerdem Threads mit einer niedrigeren Priorität verdrängen. Dadurch müssen manche Threads warten, bis Threads mit einer höheren Priorität ausgeführt worden sind. Durch das Erhöhen der Threadpriorität können Sie den Threads eine höhere Voreinstellung für die CPU-Zeit zuweisen und somit sicherstellen, dass diese nicht von anderen Threads verdrängt werden.

HINWEIS *Der vollständige Bereich für die Threadpriorität verläuft von 0 bis 31. Die Threadpriorität 0 ist für das Betriebssystem reserviert.*

Sie konfigurieren Arbeitsthreads, Fibers und die Threadpriorität im Dialogfeld **SQL Server-Eigenschaften**. Klicken Sie auf die Registerkarte **Prozessor**. Es stehen Ihnen folgende Optionen zur Verfügung:

- **Arbeitsthreadanzahl (maximal)** Legt die maximale Anzahl der Threads fest. Standardmäßig wird in diesem Feld der Wert 255 angezeigt. Sie können jedoch einen beliebigen Wert von 10 bis 32 767 festlegen (wenn Sie nicht die Desktopversion verwenden). Auf einem stark ausgelasteten Server mit zahlreichen Benutzerverbindungen können Sie diesen Wert erhöhen. Auf einem langsamen Server mit wenigen Benutzerverbindungen können Sie diesen Wert verringern. Computer mit mehreren Prozessoren können einen Thread pro CPU parallel ausführen.
- **SQL Server-Priorität unter Windows höher stufen** Erhöht die Priorität von SQL Server-Threads. Ohne Erhöhung haben die SQL Server-Threads die Priorität 7 (normale Priorität). Mit Erhöhung haben die SQL Server-Threads die Priorität 13 (hohe Priorität). Bei einem dedizierten System, auf dem nur SQL Server ausgeführt wird, kann diese Option die Leistung verbessern. Wenn der Server jedoch weitere Anwendungen ausführt, kann dadurch die Leistung dieser Anwendungen beeinträchtigt werden.
- **Windows NT-Fibers verwenden** Konfiguriert SQL Server für die Verwendung von Fibers, die direkt abgewickelt werden können. SQL Server benötigt weiterhin Threads zum Ausführen von Tasks. SQL Server weist einen Thread pro CPU zu und ordnet anschließend einen Fiber pro paralleler Benutzerverbindung bis zum Wert für die maximale Arbeitsthreadanzahl zu. Starten Sie den Server neu, damit diese Option übernommen werden kann.

TIPP *Fibers können optimal eingesetzt werden, wenn der Server mehrere CPUs verwendet und ein relativ geringes Benutzer-zu-CPU-Verhältnis hat. Bei einer Unternehmensinstallation mit 32 CPUs und 250 Benutzern wird durch Fibers beispielsweise eine erhebliche Leistungssteigerung erzielt. Wenn Sie jedoch acht CPUs und 5000 Benutzer haben, wird die Leistung mit Fibers verringert.*

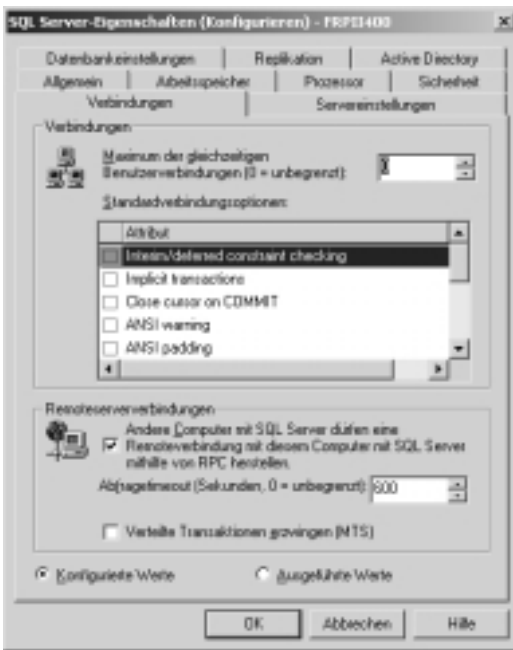
Sie können mit **sp_configure** Fibers und die maximalen Arbeitsthreads festlegen sowie die Priorität höher stufen. Es stehen hierzu folgende Befehle zur Verfügung:

```
exec sp_configure 'lightweight pooling', <0 oder 1>
exec sp_configure 'max worker threads', <Ganzzahl>
exec sp_configure 'priority boost', <0 oder 1>
```

Konfigurieren von Benutzer- und Remoteverbindungen

Anforderungen für Daten werden über Benutzerverbindungen zu Client-Systemen abgewickelt. Der Client öffnet eine Verbindung zu SQL Server, gibt eine Anfrage aus und wartet auf eine Antwort von SQL Server. Wenn der Client die Anfrage ausgeführt hat, wird die Verbindung getrennt. Andere Server und Anwendungen können auch eine Remoteverbindung zu SQL Server herstellen. Sie können im Dialogfeld **SQL Server-Eigenschaften** auf der Registerkarte **Verbindungen** Client- und Remoteserververbindungen konfigurieren.

Zu Client- und Serververbindungen gibt es zahlreiche Einstellungen (siehe Abbildung 2.8). In den folgenden Abschnitten werden diese Einstellungen beschrieben.



*Abbildung 2.8: Die Standardverbindungseinstellungen auf der Registerkarte **Verbindungen** müssen in der Regel nicht geändert werden. Notieren Sie sich die ursprünglichen Einstellungen, bevor Sie Änderungen vornehmen.*

Festlegen der maximalen Anzahl der Benutzerverbindungen

Sie können auf der Registerkarte **Verbindungen** im Feld **Maximum der gleichzeitigen Benutzerverbindungen** die maximale Anzahl der Benutzerverbindungen festlegen, die SQL Server gleichzeitig zulässt. Sie können einen beliebigen Wert im Bereich von 0 zu 32 767 festlegen. Standardmäßig wird der Wert Null (0) festgelegt. Dies bedeutet, dass eine unbegrenzte Anzahl von Verbindungen zulässig ist. Die tatsächliche Anzahl der zulässigen Benutzerverbindungen hängt jedoch von der Hardware, den Anwendungen und von anderen Servereinschränkungen ab. Sie

können die Anzahl der auf Ihrem System zulässigen Benutzerverbindungen in SQL Server Query Analyzer über folgenden Befehl bestimmen:

```
select @@max_connections
```

So legen Sie die maximale Anzahl der Benutzerverbindungen fest:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Verbindungen**.
2. Geben Sie im Feld **Maximum der gleichzeitigen Benutzerverbindungen** einen neuen Wert ein, und klicken Sie anschließend auf **OK**.
3. Starten Sie den Server neu, damit die Änderungen übernommen werden.

Sie können die Anzahl der Benutzerverbindungen auch über folgenden Befehl festlegen:

```
exec sp_configure 'user connections', <Ganzzahl>
```

PRAKTISCHE HINWEISE *In der Regel müssen Sie den Höchstwert für die Anzahl der Verbindungen nicht ändern. Sollten Sie diesen Vorgang dennoch ausführen müssen, gehen Sie bitte sorgfältig vor. Wenn die maximale Anzahl der Verbindungen erreicht wurde, erhalten die Benutzer eine Fehlermeldung und können sich erst bei der nächsten Verbindung anmelden. Dieser Wert muss nur dann festgelegt werden, wenn Ihre Umgebung viele Benutzer hat und Sie die Anzahl der aktiven Verbindungen einschränken müssen, damit die Anfragen nach Benutzerverbindungen rechtzeitig abgewickelt werden können. Eine bessere Alternative ist das Hinzufügen von ausreichendem Arbeitsspeicher im System und/oder das Konfigurieren eines Clusters zum Ausgleichen der Arbeitsauslastung. Bei einer hohen Benutzeranzahl sollten Sie außerdem sicherstellen, dass SQL-Anwendungen nach der Verbindung korrekt getrennt werden, damit weitere Ressourcen zur Verfügung stehen.*

Standardverbindungsoptionen

Auf der Registerkarte **Verbindungen** wird das Listefeld **Standardverbindungsoptionen** angezeigt. Sie können über die Optionen in dem Listefeld die Standardoptionen der Abfrageverarbeitung für Benutzerverbindungen festlegen. Zum Aus- bzw. Abwählen einer Option aktivieren bzw. deaktivieren Sie das entsprechende Kontrollkästchen. Ihre Änderungen wirken sich nur auf neue Anmeldungen aus. Die aktuellen Anmeldungen sind davon nicht betroffen. Außerdem können Benutzer den Standard gegebenenfalls mit SET-Anweisungen außer Kraft setzen.

Tabelle 2.2 enthält eine Zusammenfassung der Verbindungsoptionen und des Standardstatus für ODBC (Open Database Connectivity) und OLE DB (Object Linking and Embedding Database). Diese Vorgaben können sich vom SQL Server-Standard unterscheiden. Sie finden außerdem eine Liste der Befehle, die Sie mit **sp_configure** verwenden können, den entsprechenden Wert für die Konfigurationsbitmaske sowie die SET-Befehle zum Außerkraftsetzen der Standardeinstellungen in einer Benutzersitzung.

Tabelle 2.2: Konfigurieren der Verbindunsoptionen

Verbindungs- option	Beschreibung	Standard- status	Wert der Bitmaske	SET-Befehl
Interim/defer- red constraint checking	Wenn diese Option aktiv ist, wird die verzögerte Prüfung der FOREIGN KEY-Einschränkungen deaktiviert. Dadurch erfolgt eine strengere Prüfung der referenziellen Integrität.	Deaktiviert	1	DISABLE_DEF_CNST_CHK
Implicit trans- actions	Wenn diese Option aktiv ist, werden Transaktionen bei der Ausführung von Anweisungen implizit verwendet.	Deaktiviert	2	IMPLICIT_TRANSACTIONS
Close cursor on COMMIT	Wenn diese Option aktiv ist, wird ein Cursor am Ende einer Transaktion automatisch geschlossen.	Deaktiviert	4	CURSOR_CLOSE_ON_COMMIT
ANSI warning	Wenn diese Option aktiv ist, zeigt SQL Server NULL- und OVERFLOW-Warnungen sowie Fehler aufgrund einer Division durch NULL an. Andernfalls wird kein Fehler oder NULL zurückgegeben.	Aktiviert	8	ANSI_WARNINGS
ANSI padding	Wenn diese Option aktiv ist, werden Daten in Feldern mit fester Länge mit nachstehenden Leerzeichen aufgefüllt, um die Breite der Spalte einzuhalten.	Aktiviert	16	ANSI_PADDING
ANSI nulls	Steuert, wie NULL mit Gleichheitsoperatoren verwendet wird. Wenn diese Option aktiv ist, führen mit NULL verglichene Werte zu einem unbekannten Ergebnis.	Aktiviert	32	ANSI_NULLS



Verbindungs- option	Beschreibung	Standard- status	Wert der Bitmaske	SET-Befehl
Arithmetic abort	Wenn diese Option aktiv ist, wird eine Abfrage beendet, wenn eine OVERFLOW-Warnung oder ein Fehler aufgrund einer Division durch NULL auftritt.	Aktiviert	64	ARITHABORT
Arithmetic ignore	Gibt NULL zurück, wenn eine OVERFLOW-Warnung oder ein Fehler aufgrund einer Division durch NULL auftritt.	Aktiviert	128	ARITHIGNORE
Quoted identifier	Wenn diese Option aktiv ist, interpretiert SQL Server Angaben in Anführungszeichen nicht als Trennzeichen für eine Zeichenfolge, sondern als Bezeichner.	Aktiviert	256	QUOTED_IDENTIFIER
No count	Wenn diese Option aktiv ist, wird die Anzeige der Anzahl der in einer Abfrage zurückgegebenen Zeilen unterdrückt.	Deaktiviert	512	NOCOUNT
ANSI null defined ON	Wenn diese Option aktiv ist, werden neue Spalten so definiert, dass NULL-Werte zugelassen werden (wenn Sie nicht explizit NULL-Werte zulassen oder verweigern).	Deaktiviert	1024	ANSI_NULL_DFLT_ON
ANSI null defined OFF	Wenn diese Option aktiv ist, werden neue Spalten so definiert, dass keine NULL-Werte zugelassen werden (wenn Sie nicht explizit NULL-Werte zulassen oder verweigern).	Deaktiviert	2048	ANSI_NULL_DFLT_OFF

Für **sp_configure** werden die Standardoptionen mit dem Parameter **user options** festgelegt:

```
exec sp_configure 'user options', <Ganzzahl für den Wert der Bitmaske>
```

In diesem Beispiel ist der Wert der Bitmaske die Summe der numerischen Werte für alle Optionen, die Sie verwenden möchten. Jede Option hat auch einen entsprechenden **SET**-Befehl. Wenn Sie eine Verbindung herstellen, können Sie mit dem **SET**-Befehl die Standardeinstellung für die Sitzung außer Kraft setzen. Wenn Sie beispielsweise **ANSI padding**, **ANSI nulls** und **ANSI warning** aktivieren möchten, verwenden Sie den Bitmaskenwert 56. Beispiel:

```
exec sp_configure 'user options', 56
```

Sie können diese Optionen in einer Benutzersitzung später aktivieren oder deaktivieren.

```
set ansi_padding on  
set ansi_nulls off
```

Konfigurieren von Remoteserververbindungen

Verbindungen von anderen Servern werden anders als Benutzerverbindungen behandelt. Sie können bestimmen, ob Server die Verbindung über einen Remoteprozeduraufruf (RPC) herstellen können. Außerdem können Sie das Timeout für Remoteabfragen festlegen und bestimmen, ob verteilte Transaktionen verwendet werden. So konfigurieren Sie Remoteverbindungen:

1. Klicken Sie im Dialogfeld **Servereigenschaften** auf die Registerkarte **Verbindungen**.
2. Wenn Server die Verbindung über einen Remoteprozeduraufruf herstellen sollen, aktivieren Sie das Kontrollkästchen **Andere Computer mit SQL Server dürfen eine Remoteverbindung mit diesem Computer mit SQL Server mithilfe von RPC herstellen**. Remoteserver können sich anschließend beim Server anmelden, um gespeicherte Prozeduren über die Remoteverbindung auszuführen. Sie müssen den Server neu starten, wenn Sie diese Option ändern möchten.

ACHTUNG ***RPC-Verbindungen werden standardmäßig zugelassen. Wenn Sie dieses Verhalten ändern, können sich Remoteserver nicht mehr bei SQL Server anmelden. Dadurch wird SQL Server vor dem Remoteserverzugriff geschützt.***

3. Standardmäßig gibt es kein Timeout für Abfragen, die von Remoteservern ausgeführt werden. Um dieses Verhalten zu ändern, geben Sie im Feld **Abfragetimeout (Sekunden, 0 = unbegrenzt)** einen Wert ein. Der zulässige Bereich für diesen Wert liegt bei 0 bis 2 147 483 647 Sekunden. Der Wert Null (0) gibt an, dass für Remoteserververbindungen kein Timeout besteht.
4. Auf dem Server abgewinkelte gespeicherte Prozeduren und Abfragen können wie verteilte Transaktionen über MS DTC ausgeführt werden. Wenn Sie Prozeduren auf diese Weise ausführen möchten, aktivieren

Sie das Kontrollkästchen **Verteilte Transaktionen erzwingen (MTS)**. Sie müssen den Server neu starten, wenn Sie diese Option ändern möchten.

5. Klicken Sie auf **OK**.

Diese Optionen können auch mit **sp_configure** und den zugehörigen Transact-SQL-Anweisungen festgelegt werden:

```
exec sp_configure 'remote access', <0 oder 1>  
exec sp_configure 'remote query timeout', <Anzahl der Sekunden>  
exec sp_configure 'remote proc trans', <0 oder 1>
```

HINWEIS *Der Wert 0 bedeutet »Deaktiviert«. Mit dem Wert 1 wird die Option aktiviert.*

Servereinstellungen

Sie können im Dialogfeld **SQL Server-Eigenschaften** auf der Registerkarte **Servereinstellungen** allgemeine Einstellungen konfigurieren. Sie können unter anderem die Standardsprache und das allgemeine Serververhalten festlegen (siehe Abbildung 2.9).

Festlegen der Standardsprache für SQL Server

Die Standardsprache bestimmt das Anzeigeformat für Datumsangaben und die Namen der Monate und Tage. Wenn Sie keine lokalisierte Version von SQL Server verwenden, erfolgt die Ausgabe in Englisch (USA). Lokalisierte Versionen von SQL Server stehen in den Sprachen Französisch, Deutsch, Japanisch und Spanisch zur Verfügung. Auf einem lokalisierten Server gibt es zwei Gruppen mit Systemmeldungen, eine für Englisch, die andere für die Landessprache. Wenn die Landessprache als Standardsprache eingestellt wird, werden die SQL Server-Meldungen in der Landessprache angezeigt. Andernfalls werden Sie in Englisch (USA) ausgegeben.

Sie können im Dialogfeld **SQL Server-Eigenschaften** auf der Registerkarte **Servereinstellungen** im Dropdownlistenfeld **Standardsprache für Benutzer** die Standardsprache festlegen. Klicken Sie anschließend auf **OK**. Danach müssen Sie den Server neu starten, damit die Änderungen übernommen werden. Mit **sp_configure** lautet die zugehörige Transact-SQL-Anweisung:

```
exec sp_configure 'default language', <Sprach-ID>
```

Die Sprach-ID für U.S.-Englisch ist immer 0, für Deutsch 1031.

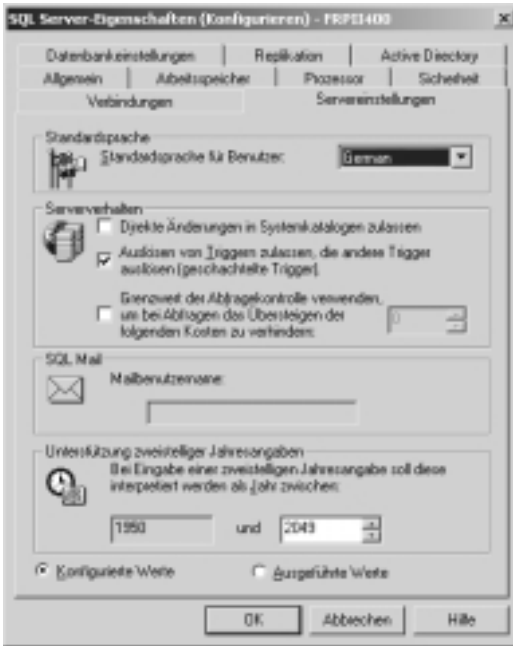


Abbildung 2.9: Auf der Registerkarte **Servereinstellungen** legen Sie allgemeine Konfigurationsoptionen für den Server fest.

Zulassen und Verweigern von Systemaktualisierungen

Standardmäßig können Benutzer die Systemtabelle nur mit systemgespeicherten Prozeduren aktualisieren. Dies gilt auch dann, wenn die Benutzer mit den entsprechenden Berechtigungen ausgestattet sind. Hierbei handelt es sich um eine wertvolle Funktion, weil dadurch verhindert wird, dass Benutzer Anweisungen ausführen, die die Datenbank beschädigen oder den Start von SQL Server blockieren. Sie können dieses Verhalten jedoch ändern und direkte Aktualisierungen der Systemtabellen zulassen. Nachdem Sie die Änderungen zugelassen haben, kann jeder Benutzer mit den geeigneten Berechtigungen Systemtabellen über Anweisungen oder gespeicherte Prozeduren aktualisieren.

Dies ist sehr riskant. Sie können stattdessen folgende Anleitung ausführen:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Servereinstellungen**.
2. Aktivieren Sie das Kontrollkästchen **Direkte Änderungen in Systemkatalogen zulassen**.
3. Klicken Sie auf **OK**, und beenden Sie SQL Server.
4. Starten Sie SQL Server im Einzelbenutzermodus. Geben Sie hierzu in der Befehlszeile **sqlservr -m** ein.

HINWEIS Wenn mehrere Instanzen installiert sind, müssen Sie hier die Option **-s (Instanzname)** verwenden, um die Instanz zu starten.

5. Nehmen Sie die erforderlichen Änderungen bei den Systemtabellen vor.
6. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Servereinstellungen**.
7. Deaktivieren Sie das Kontrollkästchen **Direkte Änderungen in Systemkatalogen zulassen**.
8. Drücken Sie in dem Fenster, in dem SQL Server ausgeführt wird, über die Tastenkombination **[STRG] + [C]**. Geben Sie anschließend den Buchstaben **Y** für »Yes« (Ja) ein. Dadurch wird SQL Server angehalten. Starten Sie den SQL Server-Dienst neu.

Mit **sp_configure** lautet die zugehörige Transact-SQL-Anweisung:

```
exec sp_configure 'allow updates', <0 oder 1>
```

HINWEIS Wenn Sie Aktualisierungen über **sp_configure** zulassen, müssen Sie auch die **RECONFIGURE WITH OVERRIDE**-Anweisung verwenden. Starten Sie anschließend den Server neu.

Zulassen und Verweigern von geschachtelten Triggern

Standardmäßig können Sie in SQL Server bis zu 32 Triggerebenen verschachteln. Geschachtelte Trigger eignen sich zum Ausführen von mehreren Tasks innerhalb einer Transaktion. Eine Aktion kann beispielsweise einen Trigger auslösen, der einen anderen Trigger startet, der ebenfalls einen Trigger startet usw. Da der Trigger innerhalb der Transaktion abgewickelt wird, hat ein Fehler auf einer beliebigen Ebene ein Rollback der gesamten Transaktion zur Folge, wodurch alle Änderungen der Datenbank rückgängig gemacht werden. Als Ausfallsicherheitsmaßnahme werden Trigger beendet, sobald die maximale Schachtelungsebene überschritten wird. Dadurch werden Schleifen verhindert.

So können Sie geschachtelte Trigger zulassen bzw. verweigern:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Servereinstellungen**.
2. Aktivieren oder deaktivieren Sie das Kontrollkästchen **Auslösen von Triggern zulassen, die andere Trigger auslösen (geschachtelte Trigger)**.
3. Klicken Sie auf **OK**.

Mit **sp_configure** lautet die zugehörige Transact-SQL-Anweisung:

```
exec sp_configure 'nested triggers', <0 oder 1>
```

Steuern der Abfrageausführung

Die Abfragekontrolle verhindert die Ausführung einer Abfrage, deren Dauer die angegebenen Kosten überschreitet. Bei den Abfragekosten handelt es sich um den geschätzten Zeitaufwand (in Sekunden) für die Abfrageausführung, und sie werden vor der Ausführung basierend auf der Analyse der Ausführungszeit des Abfragemoduls ermittelt. Standardmäßig ist

die Abfragekontrolle deaktiviert, so dass keine Kostenrechnung einbezogen wird. So aktivieren Sie die Abfragekontrolle:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Servereinstellungen**.
2. Aktivieren Sie das Kontrollkästchen **Grenzwert der Abfragekontrolle verwenden, um bei Abfragen das Übersteigen der folgenden Kosten zu verhindern**.
3. Geben Sie im Feld **Abfragekontrolle** die gewünschte Kostengrenze ein. Die Kosten liegen im Bereich zwischen 0 und 2 147 483 647. Der Wert Null (0) deaktiviert die Abfragekontrolle. Jeder andere Wert legt eine bestimmte Kostengrenze fest.
4. Klicken Sie auf **OK**.

Mit **sp_configure** lautet die zugehörige Transact-SQL-Anweisung:

```
exec sp_configure 'query governor cost limit', <Kostengrenze>
```

Sie können in Transact-SQL auch eine Obergrenze auf Einzelverbindungsbasis festlegen. Verwenden Sie hierzu folgenden Befehl:

```
set query_governor_cost_limit <Obergrenze>
```

TIPP *Bevor Sie die Abfragekontrolle verwenden, sollten Sie sich von Query Analyzer die Kosten der aktuell auf dem Server ausgeführten Abfragen berechnen lassen. Dadurch erhalten Sie eine Vorstellung von dem Wert, der für die maximalen Abfragekosten verwendet wird. Sie können mit Query Analyzer auch Abfragen optimieren.*

Festlegen des Standardprofils von SQL Mail

Sie haben zwei Möglichkeiten, um E-Mail-Programme mit SQL Server zu verwenden. Im Einzelnen sind dies folgende:

- Konfigurieren von gespeicherten Prozeduren, die durch E-Mail-Nachrichten über SQL Mail ausgelöst werden.
- Senden von Benachrichtigungen an angegebene Operatoren über SQLAgentMail.

Diese SQL Server-Funktionen können separate Mailprofile bzw. ein gemeinsames Profil verwenden. Die Profile müssen auf dem Server und für die zugehörigen gültigen E-Mail-Konten konfiguriert werden. Gehen Sie anschließend wie folgt vor:

1. Informieren Sie SQL Server über das Profil von SQL Mail. Rufen Sie hierzu das Dialogfeld **SQL Mail-Konfiguration** auf, und geben Sie anschließend im Feld **Profilname** das entsprechende Profil ein. Nachdem Sie das Mailsystem korrekt konfiguriert haben, können Sie das Dialogfeld **SQL Server-Eigenschaften** aufrufen und auf der Registerkarte **Servereinstellungen** im Feld **Mailbenutzername** einen Benutzernamen für SQL Mail eingeben.
2. Informieren Sie SQL Server über das SQLAgentMail-Profil. Rufen Sie hierzu das Dialogfeld **SQL Server-Agent-Eigenschaften** auf, und ge-

ben Sie anschließend auf der Registerkarte **Allgemein** im Feld **Mailprofil** das entsprechende Profil ein.

HINWEIS *In Kapitel 3 und Kapitel 12 finden Sie detaillierte Informationen zu SQL Mail und SQL Server-Agent.*

Konfigurieren der Unterstützung für das Jahr 2000

Sie können in SQL Server Datumsangaben ohne Angabe des Jahrhunderts einfügen oder ändern. Um jedoch die Kompatibilität mit dem Jahr 2000 zu gewährleisten, werden zweistellige Datumsangaben einer Zeitspanne zugeordnet. Standardmäßig umfasst diese Spanne die Jahre von 1950 bis 2049. Mit dieser Einstellung werden alle zweistelligen Datumsangaben von 50 bis 99 den Jahren 1950 bis 1999 und alle zweistelligen Datumsangaben von 00 bis 49 den Jahren 2000 bis 2049 zugeordnet. SQL Server interpretiert beispielsweise die Datumsangabe 99 als 1999 und die Angabe 02 als 2002.

Um die Abwärtskompatibilität beizubehalten, ändern Sie den Standardwert nicht. Sie können diesen Wert jedoch ändern. Gehen Sie hierzu folgendermaßen vor:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Servereinstellungen**.
2. Wählen Sie im Listenfeld **Bei Eingabe einer zweistelligen Jahresangabe soll diese interpretiert werden als Jahr zwischen** das letzte Jahr der jeweiligen Zeitspanne aus. Der Gültigkeitsbereich reicht von 1753 bis 9999.
3. Klicken Sie auf **OK**.

HINWEIS *Die Zeitspanne wirkt sich auf alle Datenbanken des aktuellen Servers aus. Außerdem unterstützen einige ältere OLE-Clients nur einen Datumsbereich von 1931 bis 2030. Um die Kompatibilität mit diesen Clients zu gewährleisten, können Sie 2030 als letztes Jahr der Zeitspanne eingeben.*

Mit **sp_configure** lautet die zugehörige Transact-SQL-Anweisung:

```
exec sp_configure 'two digit year cutoff', <Letztes Jahr>
```

Datenbankeinstellungen

Sie können im Dialogfeld **SQL Server-Eigenschaften** auf der Registerkarte **Datenbankeinstellungen** allgemeine Einstellungen konfigurieren. Sie können die Indexfüllung, Optionen für die Sicherung und Wiederherstellung und die Ausführung von Prüfpunkten festlegen (siehe Abbildung 2.10).

Festlegen des Füllfaktors für die Indizierung

Der Standardfaktor bestimmt, wie viel Speicherplatz von SQL Server reserviert werden soll, wenn ein neuer Index für vorhandene Daten erstellt

wird. Bei der Bestimmung des Füllfaktors muss ein Kompromiss eingegangen werden. Ein zu hoher Füllfaktor beeinträchtigt die Geschwindigkeit von SQL Server, wenn Sie Daten in eine Tabelle einfügen. Ein zu niedriger Füllfaktor kann die Lesegeschwindigkeit indirekt proportional zum Füllfaktor beeinträchtigen. Ein Füllfaktor von 25 Prozent kann beispielsweise die Lesegeschwindigkeit um den Faktor 4 (oder viermal die normale Geschwindigkeit) verringern, wobei jedoch umfangreiche Aktualisierungen zuerst schneller ausgeführt werden. Es empfiehlt sich, schnelles Aktualisieren und hohe Lesegeschwindigkeit gegeneinander abzuwägen und einen dementsprechend sinnvollen Füllfaktor zu wählen.



Abbildung 2.10: Auf der Registerkarte **Datenbankeinstellungen** legen Sie allgemeine Konfigurationsoptionen für die Datenbanken fest.

PRAKTISCHE HINWEISE *Der Füllfaktor wird nur verwendet, wenn ein Index erstellt und danach nicht verwaltet wird. Dadurch können Sie Daten einer Tabelle hinzufügen, diese Daten löschen oder aktualisieren, ohne sich Gedanken über das Verwalten eines spezifischen Füllfaktors machen zu müssen.*

Deshalb kann der freie Platz auf den Datenseiten aufgefüllt werden, wenn Sie umfangreiche Datenänderungen vornehmen. Um die Daten erneut zu verteilen, erstellen Sie den Index neu und geben dabei einen Füllfaktor an. Indizes werden in Kapitel 6 ausführlich behandelt.

Standardmäßig wird der Füllfaktor für Indizes auf Null (0) gesetzt, wobei der gültige Bereich die Werte 0 bis 100 umfasst. Die Einstellung 0 ist eine optimierte Einstellung für SQL Server. Jeder andere Wert stellt einen tatsächlichen Prozentwert dar.

SQL Server behandelt die optimierte Einstellung im Prinzip wie den Füllfaktor 100. Hier erstellt SQL Server gruppierte Indizes mit vollständigen Datenseiten und nicht gruppierte Indizes mit vollständigen Blattseiten. Im Gegensatz zu einem Füllfaktor von 100 lässt die optimierte Einstellung von 0 eine Vergrößerung auf der oberen Ebene der Indexstruktur zu. Andererseits lässt der Füllfaktor 100 keine Vergrößerung zu. Verwenden Sie diesen Wert deshalb nur bei schreibgeschützten Tabellen, in die keine neuen Daten eingefügt werden.

Sie können gegebenenfalls den Standard außer Kraft setzen, wenn Sie Indizes erstellen. Sie dürfen es nur nicht vergessen. Sie können außerdem einen festen Füllfaktor für den Index als Standard festlegen. Gehen Sie hierzu folgendermaßen vor:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Datenbankeinstellungen**.
2. Aktivieren Sie das Optionsfeld **Fest**.
3. Legen Sie mit dem Schieberegler einen Prozentwert fest. Ein geringer Füllfaktor bietet mehr Platz für Einfügungen ohne Seitenumbruch. Der Index benötigt jedoch einen höheren Speicherplatz. Ein hoher Füllfaktor bietet weniger Platz für Einfügungen ohne Seitenumbruch. Der Index benötigt jedoch einen geringeren Speicherplatz.
4. Klicken Sie auf **OK**.

Mit **sp_configure** lautet die zugehörige Transact-SQL-Anweisung:

```
exec sp_configure 'fill factor', <Prozentwert als Ganzzahl>
```

Timeout für Sicherung und Wiederherstellung

SQL Server-Sicherungen werden häufig auf Bandlaufwerken gespeichert. Wenn Sie Bandlaufwerke und die DB-Library verwenden, können Sie gegebenenfalls ein Schreib-/Lesetimeout festlegen. Nach der Standardeinstellung wartet die DB-Library »unbegrenzt lange« auf eine Antwort von SQL Server. Diese Einstellung ist in vielen Situationen alles andere als ideal. Bei einem unbegrenzten Timeout erhalten Sie in der Regel keine Fehlermeldungen, die Sie darüber informieren, dass ein Sicherungsproblem aufgetreten ist. Um dieses Verhalten zu ändern, können Sie ein bestimmtes Timeout, beispielsweise »Nach einem Versuch beenden« oder »Nach einer bestimmten Anzahl an Minuten beenden« festlegen.

So können Sie das Timeout festlegen:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Datenbankeinstellungen**.
2. Um kein Timeout festzulegen, aktivieren Sie das Optionsfeld **Unbegrenzt warten**.
3. Um nur einen einmaligen Versuch zu unternehmen, aktivieren Sie das Optionsfeld **Einmal versuchen und abbrechen**.

4. Um einen festen Zeitraum anzugeben, aktivieren Sie das Optionsfeld **Versuchsdauer** und geben Sie anschließend im entsprechenden Feld das Timeout in Minuten ein.
5. Klicken Sie auf **OK**.

Aufbewahrungsdauer für die Sicherung und Wiederherstellung

In Kapitel 11 werden die zahlreichen SQL Server-Funktionen zum Sichern und Wiederherstellen von Daten erläutert. Wenn Sie mithilfe der DB-Bibliothek Daten auf Bänder schreiben, können Sie die Anzahl der Tage angeben, die die alten Dateien aufbewahrt werden sollen. Dieser Wert wird *Aufbewahrungsdauer* genannt. So können Sie die Aufbewahrungsdauer festlegen:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Datenbankeinstellungen**.
2. Geben Sie die Anzahl der Tage, die alte Dateien aufbewahrt werden sollen, im Feld **Standardbeibehaltungsdauer für Sicherungsmedien (Tage)** ein. Wenn Sie den Mindestwert Null (0) eingeben, werden alte Dateien immer überschrieben. Der Gültigkeitsbereich beträgt 0 bis 365 Tage.
3. Klicken Sie auf **OK**.

Mit **sp_configure** lautet die zugehörige Transact-SQL-Anweisung:

```
exec sp_configure 'media retention', <Anzahl der Tage>
```

Leeren des Cache mit Prüfpunkten

Datenbankprüfpunkte leeren alle zwischengespeicherten Datenseiten auf dem Datenträger und werden pro Datenbank ausgeführt. In SQL Server können Sie steuern, wie oft Prüfpunkte über das Wiederherstellungsintervall auftreten. Standardmäßig gilt ein Wiederherstellungsintervall von Null (0). Bei dieser Einstellung kann SQL Server dynamisch steuern, wann Prüfpunkte auftreten. Dies bedeutet in der Regel, dass Prüfpunkte ungefähr einmal pro Minute auf aktiven Datenbanken auftreten. Ändern Sie diese Option nur dann, wenn bei Ihnen Probleme mit der Leistung auftreten, die mit Prüfpunkten zusammenhängen.

So können Sie das Prüfpunktintervall manuell festlegen:

1. Klicken Sie im Dialogfeld **SQL Server-Eigenschaften** auf die Registerkarte **Datenbankeinstellungen**.
2. Geben Sie im Feld **Wiederherstellungsintervall (Minuten)** das gewünschte Intervall ein. Der Gültigkeitsbereich beträgt 0 bis 32 767 Minuten. Hierbei handelt es sich um eine serverweite Einstellung.
3. Klicken Sie auf **OK**.

Mit **sp_configure** lautet die zugehörige Transact-SQL-Anweisung:

```
exec sp_configure 'recovery interval', <Anzahl der Minuten>
```

Replikation

Sie können im Dialogfeld **SQL Server-Eigenschaften** auf der Registerkarte **Replikation** Publizierungs- und Verteilungsdienste für SQL Server verwalten und konfigurieren. In diesem Abschnitt werden die verfügbaren Optionen auf der Registerkarte **Replikation** erläutert. Außerdem erfahren Sie, wie der Publizierungs- und Verteilungskonfigurations-Assistent gestartet wird.

Verteilung und Publizierung

Zuerst wird auf der Registerkarte **Replikation** im Abschnitt **Publizierung und Verteilung** der Eintrag **COMPUTERNAME\SQLINSTANZ ist nicht als Verleger konfiguriert** angezeigt. Um die Dienste für die Publizierung und Verteilung zu konfigurieren, klicken Sie auf **Konfigurieren**, um den Publizierungs- und Verteilungskonfigurations-Assistenten aufzurufen. Sie können mit diesem Assistenten folgende Aktionen ausführen:

- Sie können die lokale SQL Server-Instanz oder einen anderen Server als Verteiler angeben.
- Sie können die Eigenschaften der lokalen SQL Server-Instanz als Verteiler konfigurieren.
- Sie können die Eigenschaften der lokalen SQL Server-Instanz als Verleger konfigurieren.

Wenn Sie die lokale SQL Server-Instanz als Verleger und Verteiler konfigurieren, wird im Bereich **Publizierung und Verteilung** der Eintrag **Konfiguration von COMPUTERNAME\SQLINSTANZ als Verteiler anzeigen oder ändern** angezeigt. Wenn Sie auf **Konfigurieren** klicken, wird die Seite **Verleger- und Verteilereigenschaften** für den lokalen SQL Server aufgerufen. Sie können auf der Seite **Verleger- und Verteilereigenschaften** die Eigenschaften für Verteiler, Verleger, Publikationsdatenbanken und Abonnenten aufrufen und ändern.

Wenn Sie die lokale SQL Server-Instanz als Verleger konfigurieren, wird im Bereich **Publizierung und Verteilung** der Eintrag **Konfiguration von COMPUTERNAME\SQLINSTANZ als Verleger anzeigen oder ändern** angezeigt. Wenn Sie auf **Konfigurieren** klicken, wird die Seite **Verleger- und Verteilereigenschaften** aufgerufen. Sie können jedoch nur auf die Eigenschaften für Verteiler, Publikationsdatenbanken und Abonnenten zugreifen.

Wenn Sie auf **Deaktivieren** klicken, wird der Begrüßungsbildschirm des Publizierungs- und Verteilungsdeaktivierungs-Assistenten aufgerufen. Sie können mit diesem Assistenten die Publizierung und/oder Verteilung auf dem lokalen SQL Server deaktivieren.

Replikationsmonitorgruppe

Auf der Registerkarte **Replikation** wird ein neuer Abschnitt mit der Bezeichnung **Replikationsmonitorgruppe** angezeigt. Wenn Sie das Kon-

trollkästchen **COMPUTERNAME\SQLINSTANZ als Verteiler zur Replikationsmonitorgruppe hinzufügen** aktivieren, wird der lokale SQL Server der Replikationsmonitorgruppe als Verteiler hinzugefügt. Sie können dann die Replikationsaktivität der Verteiler vom lokalen Computer aus überwachen.

Weitere Informationen zu den Replikationsmodi sowie zum Einrichten und Konfigurieren von Publizierungs- und Verteilungsservern für Abonnenten finden Sie in Kapitel 9, »Konfigurieren von Snapshot-, Merge- und Transaktionsreplikationen«.

Hinzufügen und Entfernen von Active Directory-Informationen

Sie können im Dialogfeld **SQL Server-Eigenschaften** auf der Registerkarte **Active Directory** SQL Server-Informationen verwalten, die in Active Directory-Diensten publiziert werden. Die Registerkarte enthält drei Schaltflächen:

- **Hinzufügen** Publiziert Informationen zu einer SQL Server-Instanz in Active Directory.
- **Aktualisieren** Aktualisiert Informationen zu einer SQL Server-Instanz in Active Directory. Diese Option ist nützlich, wenn Sie Datenbanken, Servercubes oder Dataminingmodelle erstellen und die Aktualisierungen vor der normalen Replikation im Verzeichnis speichern möchten.
- **Entfernen** Entfernt Informationen zu einer SQL Server-Instanz aus Active Directory.

Konfigurieren von SQL Server mit gespeicherten Prozeduren

Sie können zahlreiche Bereiche von SQL Server über das Dialogfeld **SQL Server-Eigenschaften** konfigurieren. Wie Sie bereits in diesem Kapitel erfahren haben, können Sie SQL Server auch mit gespeicherten Prozeduren konfigurieren, beispielsweise mit **sp_configure**. Sie führen gespeicherte Prozeduren und andere Abfragen in Query Analyzer aus. Query Analyzer ist ein Clienttool, das Befehle an das SQL Server-Abfragemodul sendet, das die Befehle analysiert, kompiliert und ausführt.

In den folgenden Abschnitten wird erläutert, wie Sie SQL Server mit Query Analyzer und gespeicherten Prozeduren konfigurieren können. In anderen Kapiteln finden Sie detaillierte Erläuterungen zu Query Analyzer.

Starten von Query Analyzer

Sie haben verschiedene Möglichkeiten, um Analyzer zu starten:

- Wählen Sie **Start**, danach **Programme**, **Microsoft SQL Server** und anschließend **Query Analyzer**.

- Führen Sie **ISQLW.EXE** in der Befehlszeile aus.
- Klicken Sie im Menü **Extras** von Enterprise Manager auf **SQL Query Analyzer**.

Wenn Sie Query Analyzer über das Menü oder die Befehlszeile öffnen, müssen Sie in einem Dialogfeld Verbindungsinformationen angeben (siehe Abbildung 2.11). Sie können die Felder in diesem Dialogfeld folgendermaßen verwenden:

- **SQL Server** Verwenden Sie diese Option, um die Datenbank-Serverinstanz auszuwählen, zu der eine Verbindung hergestellt werden soll. Wenn die Serverinstanz im Dropdownlistenfeld nicht angezeigt wird, können Sie einen Servernamen eingeben oder auf die Schaltfläche (...) klicken, um das Dialogfeld **Server auswählen** aufzurufen, in dem Sie alle SQL Server innerhalb der vollständigen Active Directory-Gesamtstruktur durchsuchen können.
- **SQL Server starten, falls SQL Server beendet wurde** Startet den MS-SQLServer-Dienst und die Datenbank (falls diese beendet wurde).
- **Windows-Authentifizierung** Verwendet Ihr aktuelles Domänenkonto und Kennwort, um die Datenbankverbindung einzurichten. Dieser Vorgang kann nur ausgeführt werden, wenn die Windows-Authentifizierung aktiviert ist und Sie die geeigneten Zugriffsberechtigungen besitzen.
- **SQL Server-Authentifizierung** Ermöglicht die Angabe eines SQL Server-Benutzernamens und des Kennworts.

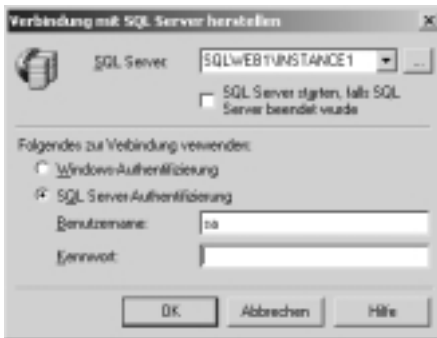


Abbildung 2.11: Sie können im Dialogfeld **Verbindung mit SQL Server herstellen** den gewünschten Server auswählen und anschließend die Verbindungsinformationen eingeben. Wenn Sie **Query Analyzer** im **Enterprise Manager** starten, können Sie dieses Dialogfeld oft umgehen.

Wenn Sie in Enterprise Manager mit einer aktiven Datenbank arbeiten und die Verbindung bereits authentifiziert haben, stellt Query Analyzer automatisch eine Verbindung zur aktuell ausgewählten Datenbank-Serverinstanz her und verwendet diese Authentifizierungsinformationen für die Anmeldung. Wenn keine aktive Datenbankverbindung vorhanden ist oder wenn die Anmeldung fehlschlägt, wird ein Dialogfeld angezeigt (siehe Abbildung 2.11).

TIPP Query Analyzer stellt in der Regel eine Verbindung zur master-Datenbank auf dem aktiven Server her. Sie können im Dropdownlis-

tenfeld DB auf der Query Analyzer-Symbolleiste die verfügbaren Datenbanken auf dem Server ändern.

Ändern von Einstellungen mit `sp_configure`

Das Fenster von Query Analyzer ist in der Regel in drei Bereiche unterteilt (siehe Abbildung 2.12). Im linken Bereich können Sie die Objekte durchsuchen, die auf der aktuell ausgewählten Datenbank verfügbar sind. Der obere rechte Bereich ist der Eingabebereich für Abfragen. Der untere rechte Bereich ist der Anzeigebereich für die Ergebnisse.

Wenn der untere rechte Bereich nicht angezeigt wird, ist das kein Problem. Dieser Bereich wird automatisch angezeigt, wenn Sie eine Abfrage eingeben. Sie können auch festlegen, dass dieser Bereich standardmäßig angezeigt wird. Wählen Sie hierzu im Menü **Fenster** den Befehl **Ergebnisbereich anzeigen**.

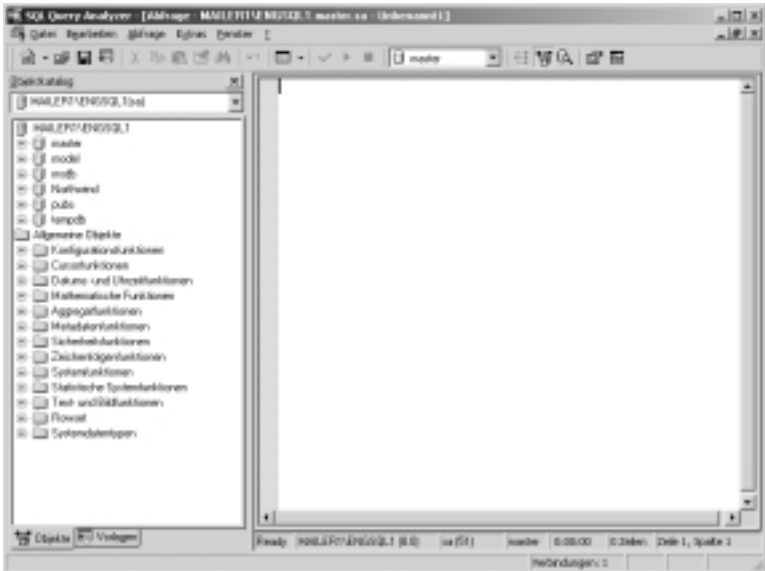


Abbildung 2.12: Das neueste Feature von Query Analyzer ist der Objektkatalog. Sie können in diesem Bereich unter anderem zu Datenbanken, Tabellen und Funktionen navigieren.

Sie können mit `sp_configure` die SQL Server-Konfigurationseinstellungen anzeigen und ändern. Es stehen zwei Arten von Konfigurationseinstellungen zur Verfügung: dynamische und statische. In dieser Instanz können Sie eine dynamische Einstellung ohne Neustart von SQL Server ändern. Um `sp_configure` oder andere Abfragetypen auszuführen, geben Sie im oberen Bereich einen Befehl ein und klicken anschließend auf die Schaltfläche **Abfrage ausführen** in der Symbolleiste (der grüne Pfeil). Sie können Befehle auch über folgende Tastenkombinationen ausführen:

- F5
- STRG + E
- ALT + X

Wenn Sie Einstellungen über **sp_configure** ändern, werden die Änderungen erst übernommen, wenn Sie auch den Befehl RECONFIGURE ausführen. Sie können einige äußerst riskante Einstellungen nur mit dem Befehl RECONFIGURE WITH OVERRIDE vornehmen. Außerdem werden die Einstellungen von **sp_configure** in zwei Kategorien unterteilt: **Standard** und **Erweitert**. Sie können Standardbefehle jederzeit ausführen, erweiterte Befehle können nur dann angezeigt werden, wenn **show advanced options** auf 1 gesetzt ist. Wenn diese Einstellung aktiviert ist, können Sie sowohl Standard- als auch erweiterte Einstellungen über folgende Prozedur ändern:

1. Geben Sie in Query Analyzer Folgendes ein:

```
exec sp_configure 'show advanced options', 1
go
reconfigure
go
```

HINWEIS Sie können die erweiterten Optionen später deaktivieren, indem Sie diesen Wert auf 0 setzen.

2. Führen Sie die Befehle über STRG + E aus.
3. Löschen Sie den Inhalt der Abfragefenster, indem Sie auf der Symbolleiste auf die Schaltfläche **Abfragefenster löschen** klicken.
4. Geben Sie jetzt für jede zu ändernde Option den Befehl **sp_configure** ein.
5. Geben Sie **reconfigure** (oder **reconfigure with override**) ein.
6. Führen Sie die Befehle über STRG + E aus.
7. Wenn Sie nichtdynamische Einstellungen geändert haben, halten Sie den Server an, und starten Sie ihn neu. (Weitere Informationen hierzu finden Sie in den Tabellen 2.3 und 2.4.)

Konfigurationsparameter

Tabelle 2.3 enthält eine Zusammenfassung der Standardkonfigurationsparameter. Die Parameter werden alphabetisch mit den Mindest-, Höchst- und Standardwerten angezeigt. In der Spalte für den dynamischen Parameter erfahren Sie jeweils, ob es sich um eine dynamische Einstellung handelt. Wenn in dieser Spalte »No« eingetragen ist, müssen Sie beim Server einen Neustart ausführen, bevor die Änderungen in Kraft treten.

Tabelle 2.3: Zusammenfassung der Standardkonfigurationsparameter

Parameter- name	Beschreibung	Mindest- wert	Höchst- wert	Standard- wert	Dynamisch (Ja/Nein)
allow updates	Ermöglicht Aktualisierungen von Systemtabellen.	0	1	0	Ja
default language	Legt das Sprachformat für Datumsangaben fest. Kann bei lokalisierten Versionen die Landessprache oder amerikanisches Englisch sein.	0	9999	Unterschiedlich, 0 ist U.S.- Englisch, Deutsch 1031	Ja
max text repl size	Legt die maximale Anzahl von Bytes fest, die in eine replizierte Spalte geschrieben werden können.	0	2147483647	65536	Ja
nested triggers	Bestimmt, ob geschachtelte Trigger zulässig sind.	0	1	1	Ja
remote access	Steuert den Zugriff von Remoteservern. 0 verweigert den Zugriff.	0	1	1	Nein
remote login timeout	Legt das Anmeldezeitout für Remoteanmeldungen (in Sekunden) fest. 0 setzt kein Timeout.	0	2147483647	5	Ja
remote proc trans	Verwendet DTC für verteilte Transaktionen.	0	1	0	Ja
remote query timeout	Setzt das Timeout (in Sekunden) für Abfragen von Remoteservern. 0 setzt kein Timeout.	0	2147483647	0	Ja
show advanced options	Wenn Sie den Parameter auf 1 setzen, werden die erweiterten Optionen aktiviert.	0	1	1	Ja
user options	Setzt einen globalen Standard für Benutzerverbindungen mit einer Bitmaske.	0	16383	0	Ja

Tabelle 2.4 enthält eine Zusammenfassung der erweiterten Konfigurationsparameter. Um diese Parameter anzuzeigen oder zu ändern, müssen Sie den Parameter **show advanced options** auf 1 setzen. Beachten Sie, dass Sie keine erweiterten Optionen ändern können. (Sie können diese lediglich anzeigen.)

Tabelle 2.4: Zusammenfassung der erweiterten Konfigurationsparameter

Parameter- name	Beschreibung	Mindest- wert	Höchstwert	Standard- wert	Dynamisch (Ja/Nein)
affinity mask	Ordnet Threads Prozessoren zu.	0	2147483647	0	Nein
AWE enabled	Aktiviert die Unterstützung für sehr große Adressräume.	0	1	0	Ja
cost thresh- hold for parallelism	Hilft bei der Bestimmung der parallelen Verarbeitung der Abfragen.	0	32767	5	Ja
cursor threshold	-1 zum Generieren von synchronen Schlüsselsätzen für den Cursor. 0 setzt auf asynchron. Die anderen Werte führen einen Zeilenvergleich aus.	-1	2147483647	-1	Ja
default full-text language	Legt die Standardsprache für die Volltextsuche in der Datenbank fest.	0	2147483647	0	Ja
default language	Legt die Standardsprache in der Datenbank fest.	0	9999	0	Ja
fill factor	Legt den Prozentwert für den Indexfüllfaktor fest. 0 lässt SQL Server konfigurieren.	0	100	0	Nein
index create memory	Legt die KB für den Arbeitsspeicher zum Erstellen von Indizes für das Sortieren von Daten fest.	704	2147483647	0	Ja
lightweight pooling	1 wechselt zu Fibers.	0	1	0	Ja
locks	Legt die Anzahl der verfügbaren Sperren fest.	5000	2147483647	0	Nein



Parameter- name	Beschreibung	Mindest- wert	Höchstwert	Standard- wert	Dynamisch (Ja/Nein)
max degree of parallelism	Legt mit SMP die Anzahl der Threads für die Ausführung eines parallelen Planes fest.	0	32	0	Ja
max server memory	Legt den maximalen Arbeitsspeicher (MB) für SQL Server fest.	4	2147483647	Unterschiedlich	Ja
max worker threads	Legt die maximale Anzahl der Threads fest.	10	32767	255	Ja
media retention	Legt die Anzahl der Tage fest, für die Sicherungsmedien beibehalten werden.	0	365	0	Nein
min memory per query	Legt den Mindestwert für den Arbeitsspeicher (in KB) fest, der für Abfragen reserviert wird.	512	2147483647	1024	Ja
min server memory	Legt den minimalen Arbeitsspeicher (MB) für SQL Server fest.	0	2147483647	0	Ja
network packet size	Legt die Bytegröße von Netzwerkpaketen fest.	512	65535	4096	Ja
open objects	Legt die maximale Anzahl der offenen Objekte fest. 0 für SQL-Steuerung.	0	2147483647	0	Nein
priority boost	Wenn Sie den Wert 1 festlegen, wird die Thread-Priorität höher gestuft.	0	1	0	Nein
query governor cost limit	Legt die Kosten- grenze für Abfragen fest. 0 deaktiviert die Grenze.	0	2147483647	0	Nein
query wait	Legt das Timeout für Abfragen fest.	-1	2147483647	-1	Ja
recovery interval	Legt das Zeitintervall (in Minuten) für Prüfpunkte fest. 0 für SQL-Steuerung.	0	32767	0	Ja



Parameter-name	Beschreibung	Mindestwert	Höchstwert	Standardwert	Dynamisch (Ja/Nein)
scan for startup procs	Legen Sie den Wert 1 fest, damit die gespeicherte Prozedur ermittelt wird, die beim Start ausgeführt werden soll.	0	1	0	Nein
set working set size	Der Wert 1 deaktiviert die Auslagerung für den Arbeitsspeicher.	0	1	0	Nein
two digit year cutoff	Legt den Datumsbereich für zweistellige Datumsangaben fest.	1753	9999	2049	Ja
user connections	Steuert die maximale Anzahl der Benutzerverbindungen für die Datenbank-Serverinstanz. Der Wert 0 legt eine unbegrenzte Anzahl fest.	0	32767	0	Nein

Problembehandlung bei Konfigurationsproblemen

Die folgenden Abschnitte untersuchen zwei bestimmte Techniken zum Lösen von SQL Server-Konfigurationsproblemen. Sie erfahren, wie Sie eine ungültige Konfiguration wiederherstellen und die **master**-Datenbank erneut erstellen.

Wiederherstellen einer gültigen Konfiguration

SQL Server 2000 bietet viele Schutzfunktionen, die gewährleisten, dass keine Konfigurationseinstellungen vorgenommen werden können, die den Start von SQL Server verhindern. Es ist dennoch möglich, dass SQL Server durch eine Konfigurationsänderung nicht mehr gestartet werden kann. So können Sie eine Serverinstanz gegebenenfalls wiederherstellen:

1. Melden Sie sich bei dem betroffenen Server lokal oder über Telnet oder Terminal Server an. Sie müssen sich über ein lokales Administratorkonto oder über das Konto anmelden, das von der Datenbank-Serverinstanz verwendet wird.
2. Vergewissern Sie sich, dass der Dienst MSSQLServer bzw. MSSQL\$*instancename* angehalten ist. Halten Sie gegebenenfalls den Dienst mit einer der folgenden Methoden an:
 - SQL Server-Dienst-Manager

- Windows NT-Dienst-Manager
- Windows 2000-Dienste

3. Wenn die Instanz von SQL Server als Standardinstallation eingerichtet wurde, können Sie den Dienst über folgenden Befehl anhalten:

```
net stop MSSQLSERVER
```

4. Rufen Sie über die Eingabeaufforderung das Verzeichnis der zugehörigen SQL Server-Instanz (entweder **mssql\binn** oder **mssql\$instance-name\binn**) auf. Sie müssen sich in diesem Verzeichnis befinden, um das Dienstprogramm sqlservr zu verwenden.

5. Starten Sie SQL Server über die Befehlszeile mit folgender Option:

```
sqlservr -s(instance_name) -f
```

6. Sie müssen über die Option -s die SQL Server-Instanz angeben, wenn mehrere Instanzen von SQL Server installiert sind. Die Option -f startet SQL Server im Einzelbenutzermodus mit einem minimalen Konfigurationsaufwand. Dadurch wird sichergestellt, dass die ungültige Konfiguration nicht geladen wird.

7. Warten Sie auf den Start von SQL Server. SQL Server gibt einige Seiten am Bildschirm aus. Warten Sie, bis dieser Vorgang beendet ist.

8. Starten Sie in einer anderen Eingabeaufforderung oder Telnet-Sitzung ISQL mit dem Benutzernamen eines SQL-Kontos mit Administratorrechten und einem Kennwort:

```
isql -U Benutzername -P Kennwort
```

HINWEIS *Sie müssen die Instanz angeben, zu der Sie die Verbindung herstellen (isql, -U Benutzername, -P Kennwort, -S computer-name\instanzname), wenn mehrere SQL Server 2000-Instanzen installiert sind.*

9. Wenn Sie ISQL korrekt aufgerufen haben, wird in der Eingabeaufforderung > angezeigt.

10. Kehren Sie die vorherigen Konfigurationsänderungen um, indem Sie Befehle eingeben, die mit den Befehlen von Query Analyzer vergleichbar sind. Der Unterschied liegt in erster Linie darin, dass die Befehle mit der Zeichenfolge go bestätigt werden. Beispiel:

```
exec sp_configure 'max server memory', 128
go
reconfigure
go
```

11. Verlassen Sie anschließend ISQL über den Befehl **exit**.

12. Drücken Sie **[STRG]+[C]** in dem Fenster, in dem SQL Server über die Befehlszeile ausgeführt wird. Geben Sie anschließend den Buchstaben **Y** für »Yes« (Ja) ein. Dadurch wird SQL Server angehalten.

13. Starten Sie jetzt SQL Server wie gewöhnlich. Wenn Sie die korrekten Änderungen vorgenommen haben, wird der Server normal gestartet. Sollte dies nicht der Fall sein, führen Sie die obigen Schritte erneut durch.

Ändern der Sortierung und Neuerstellen der **master**-Datenbank

Für das erneute Erstellen der **master**-Datenbank gibt es folgende Hauptgründe

- Sie müssen eine neue Standardsortierung für eine Datenbank-Serverinstanz festlegen.
- Sie müssen eine beschädigte **master**-Datenbank reparieren.

Nachdem Sie die **master**-Datenbank neu erstellt haben, werden alle Benutzerdatenbanken getrennt und können nicht gelesen werden. Um diese wiederherzustellen, müssen Sie alle Benutzerdatenbanken neu erstellen. Sie können die Benutzerdatenbanken nicht aus einer Sicherung wiederherstellen: Die Wiederherstellung behält die Informationen bei, die beim Erstellen der Sicherung festgelegt wurden. Sie können stattdessen die Datenbanken über DTS (Data Transformation Services) zu einem anderen Server verschieben. Weitere Informationen zu DTS finden Sie in Kapitel 7.

HINWEIS *Bevor Sie die master-Datenbank erneut erstellen, müssen Sie beachten, dass die Sortierung bei SQL Server 2000 anders als bei SQL Server 7 abgewickelt wird. Frühere Versionen von SQL Server haben für alle Datenbanken und Datenbankobjekte einer Instanz immer die gleiche Sortierung verwendet. SQL Server 2000 kann unterschiedliche Sortierungen für separate Datenbanken und Spalten innerhalb einer Datenbank angeben. Wenn Sie SQL Server installieren, wird die Standardsortierung festgelegt. Sie können diese Standardeinstellung bei einer vorhandenen SQL Server-Installation nur ändern, indem Sie die master-Datenbank neu erstellen. Sie können jedoch separate Sortierungsoptionen für jede Datenbank festlegen, die Sie auf dem Server installieren. Sie können auch separate Sortierungsoptionen für Tabellen, Parameter und Literalzeichenfolgen festlegen.*

So können Sie die **master**-Datenbank neu erstellen:

1. Halten Sie den SQL Server-Dienst für die Datenbank-Serverinstanz an, mit der Sie arbeiten möchten. Die Standardinstanz dieses Dienstes heißt MSSQLServer. Bei benannten Instanzen heißt der Dienst MSSQL\$*instancename*.

TIPP *Weitere Informationen zum Anhalten des SQL Server-Dienstes finden Sie in Kapitel 3 unter »Starten, Beenden und Anhalten von SQL Server«.*

2. Führen Sie *Rebuild Master* über die Befehlszeile mit folgendem Befehl aus:

```
rebuilddm
```

3. Das Dialogfeld **Rebuild Master** wird angezeigt. Wählen Sie aus der Auswahlliste **Server** die Datenbank-Serverinstanz aus, mit der Sie arbeiten möchten.
4. Im Feld **Quellverzeichnis, das Datendateien enthält** wird der ursprüngliche Speicherort für die Datenbank-Serverinstanz angezeigt. In

der Regel bezieht sich das Quellverzeichnis auf einen Speicherort auf der CD-ROM für SQL Server. Sie können gegebenenfalls einen neuen Verzeichnispfad eingeben oder auf **Durchsuchen** klicken, um zu dem Verzeichnispfad zu navigieren.

5. Klicken Sie auf **Einstellungen**, und legen Sie anschließend im Dialogfeld **Sortierungseinstellungen** die neuen Servereinstellungen fest. Klicken Sie auf **OK**.
6. Klicken Sie im Dialogfeld **Rebuild Master** auf **Erneut erstellen**. Wenn Sie aufgefordert werden, diesen Schritt zu bestätigen, klicken Sie auf **Ja**. Rebuild Master kopiert die Quelldateien für die **master**-Datenbank auf den Server und beginnt anschließend mit der neuen Zuordnung und Konfiguration des Servers.