

4

Управление механизмом баз данных и исправление ошибок

После установки сервера основной задачей, стоящей перед администратором, является настройка его рабочей среды. В этой главе речь пойдет о процедуре настройки экземпляра SQL Server. После настройки экземпляра мы узнаем, как работать с экземпляром с помощью утилиты Management Studio, группы хранимых процедур или DMV-курсов. Эта глава также расскажет об управлении подключениями к SQL Server и о том, как решать возникающие проблемы.

Здесь предполагается, что вы уже знакомы с основами работы среды Management Studio, и упор делается, прежде всего, на том, что должен знать и уметь DBA. В других главах этой книги подробно будет рассказано о том, как эксплуатировать различные возможности Management Studio, а в этой главе эти моменты просто опущены (о резервном копировании, например, речь пойдет в главе 18 “Резервное копирование и восстановление”).

Средства настройки конфигурации

После инсталляции SQL Server или перехода со старой версии на SQL Server 2005, он, скорее всего, не будет иметь всех необходимых вам настроек. При разработке SQL Server 2005 компания Microsoft выбрала путь существенного ограничения набора активизированных возможностей SQL Server, отключая во время инсталляции максимум возможностей. Отключение возможностей зависит от версии SQL Server. Например, протокол TCP/IP отключен в версии Developer по умолчанию. И интеграция CLR тоже отключена по умолчанию. Это делает среду более удобной для администратора, так как в этом случае ненужные возможности не переполняют ваш экран. При этом также снижается вероятность проникновения хакеров в вашу систему.

Диспетчер настройки SQL Server

Диспетчер настройки SQL Server (SQL Server Configuration Manager) позволяет проводить настройку служб SQL Server во многом аналогично апплету Службы из Панели управления, предоставляя при этом гораздо большую функциональность, чем этот апплет. Например, диспетчер настройки позволяет вносить изменения в установки прослушиваемых SQL Server портов и используемого каждым экземпляром порта. Открыть программу можно следующим образом: Пуск⇒SQL Server 2005⇒Configuration Tools (рис. 4.1).

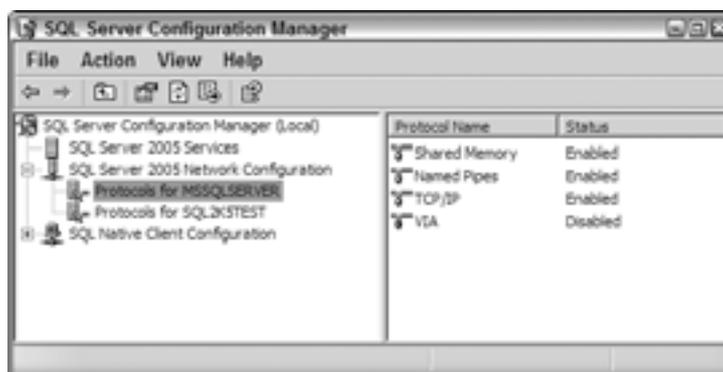


Рис. 4.1. Диалоговое окно диспетчера настройки SQL Server Configuration Manager

Настройку служб SQL Server можно производить из дерева SQL Server 2005. Для настройки таких отдельных служб, как SQL Server, двойным щелчком на имени службы откройте страницу Properties. На вкладке Log On вы сможете задать учетную запись, под управлением которой будет запускаться SQL Server. Его рекомендуется запускать под управлением учетной записи, обладающей минимальными правами. Например, эта учетная запись должна иметь право локальной регистрации. Этой учетной записи необязательно быть локальным администратором или администратором домена в SQL Server 2005. Кроме того, такой учетной записи необходимо задавать бессрочный пароль, чтобы не возникла ситуация, когда SQL Server вдруг неожиданно не сможет запуститься из-за завершения срока действия пароля используемой им учетной записи. Когда службе SQL Server нет необходимости общаться с внешним миром, службу следует запускать под управлением локальной системной учетной записи, но эта учетная запись может иметь больше локальных прав, чем это требуется на самом деле (подробнее об этом — в главе 9 “Безопасность механизма баз данных”).

Во вкладке Service можно задать режим запуска службы: вручную, автоматически или отключить ее совсем. Перейдя к вкладке Advanced (рис. 4.2), мы сможем задать для каждой службы и более специфические параметры, например, можно отключить возможность Customer Feedback Reporting. Эта возможность позволяет компании Microsoft получать отчеты о работе вашего SQL Server. Даже если это необходимо, в большинстве рабочих сред ваш SQL Server не будет иметь возможности отправки отчетов из-за отсутствия доступа к Интернет.

Параметр Error Reporting, имеющийся на вкладке Advanced, позволяет отправлять электронную почту в компанию Microsoft при возникновении ошибки. Самая минимальная информация отправляется по защищенному протоколу HTTPS.

С другой стороны, информацию об этих ошибках можно отсылать на внутреннюю корпоративную систему отчетов об ошибках (Corporate Error Reporting, CER), которую можно найти на Web-узле Microsoft, откуда ее можно загрузить.

На странице SQL Server 2005 Network Configuration, посвященной настройкам сети в диспетчере настройки Configuration Manager, представлен список сетевых протоколов, которые прослушивает данный экземпляр SQL Server. Если необходимо отключить какой-то протокол, то это можно сделать, нажав на правую клавишу мыши на выбранном протоколе, выбрав затем Enable или Disable. Активизацией только протокола общей памяти (Shared Memory) отключается любой обмен данными экземпляра с внешним миром.



Рис. 4.2. Диалоговое окно свойств SQL Server

Инструментальное средство Surface Area Configuration

Новый SQL Server 2005 содержит много новых возможностей, считающихся спорными среди DBA. Например, многие DBA были напуганы (и можно сказать, что серьезно) возможностью CLR-интеграции и не хотят, чтобы эта возможность включалась по умолчанию. Для того чтобы система была защищена и более работоспособна, Microsoft приняла решение отключить многие возможности по умолчанию. Средство настройки SQL Server Surface Area Configuration (сокращенно SQL SAC) позволяет активизировать возможности, отключенные по умолчанию в SQL Server и в его вспомогательных продуктах. После открытия этого средства можно произвести выбор настраиваемых параметров: службы и соединения (страница Services and Connections) или возможности (страница Features). Страница Services and Connections позволяет настраивать службы и сетевые возможности SQL Server во многом аналогично диспетчеру SQL Server Configuration Manager. А страница Features позволяет активизировать такие возможности, как SQL CLR-интеграцию.

Для выбора средства настройки на удаленном сервере можно выбрать Change Computer, если на это есть соответствующие права. При выборе Services and Connections

можно увидеть часть инструментального средства, которое по своим функциональным возможностям очень напоминает диспетчер SQL Server Configuration Manager. Например, в основном это инструментальное средство является интерфейсом, позволяющим настроить запуск служб SQL Server (автоматический запуск, запуск вручную или отключено).

Самой важной областью настройки является часть, расположенная в ветке Database Engine⇒Remote Connections (рис. 4.3). Здесь задается возможность удаленных подключений к SQL Server, в противном случае будут разрешены только ограниченные соединения в пределах вашей рабочей станции или сервера. Такая установка задается выбором установки Local Connections Only в версиях SQL Server Developer, Evaluation и SQL Express. Такая установка может потребоваться при изучении возможностей SQL Server, так как эти версии используются только для локального применения и разработки программного обеспечения, когда внешние соединения вряд-ли необходимы. В случае выбора установки Local and Remote Connections задайте протоколы, которые будет прослушивать SQL Server. Эта установка также дублируется в диспетчере SQL Server Configuration Manager. Имеются аналогичные установки и для служб Analysis Services.



Рис. 4.3. Установка удаленных соединений

Изменение этой установки потребует перезапуска SQL Server.

Основные функциональные возможности этого приложения находятся в области Configuration for Features, в которую можно попасть с главной страницы, щелкнув на ссылке Surface Area Configuration for Features. Для настройки служб с помощью этого приложения должны быть запущены службы SQL Server, а для управления экземпляром SQL Server 2005 достаточно только этого инструментального средства. Для указания режима сортировки достаточно произвести выбор между View by Instance или View by Component.

Страница Ad-Hoc Remote Queries позволяет активизировать функции OpenRowSet и OpenDataSource. Эти функции позволяют подключиться к SQL Server, а затем использовать этот сервер для подключения к другим серверам для запуска запросов на них. Эти две функции отключены по умолчанию, так как ими может воспользо-

ваться хакер для проникновения в систему. Например, хакер может воспользоваться возможностью, предоставляемой SQL, для того, чтобы подключиться к SQL Server через Web-сервер и атаковать другие серверы посредством разрешенного соединения. Эти установки необходимо держать в отключенном состоянии, если нет настоятельной необходимости с ними работать.

Активизировать одну из наиболее спорных возможностей администратора — интеграцию CLR — можно на странице CLR Integration. После ее отключения разработчики не могут использовать эту одну из наиболее революционных возможностей, имеющейся в SQL Server 2005. При такой установке, если пользователь попробует воспользоваться CLR-интеграцией, он получит сообщение об ошибке.

Страница DAC предназначена для включения выделенного подключения администратора (Dedicated Administrator Connection, DAC). В сущности, такое подключение позволяет администратору получить доступ к SQL Server, когда процессор полностью загружен и обычные пользователи не имеют возможности ничего делать. Такому подключению предоставляется экстр-приоритетность, поэтому Microsoft страшется от того, чтобы администраторы использовали это подключение в повседневной работе и отключили его.

Пользоваться DAC-подключением можно только в исключительных случаях. Например, когда видно, что какой-то пользователь запустил запрос, захвативший все ресурсы системы. Этот пользователь может забрать все системные ресурсы, не позволяя тем самым никому, включая и вас, войти в систему, чтобы разорвать соединение этого пользователя. Вот тут и можно воспользоваться DAC-подключением. При аналогичных обстоятельствах в ранних версиях SQL Server приходилось останавливать и запускать SQL Server по прошествию определенного времени. Возможностью DAC-подключения можно воспользоваться, если вы обладаете правами системного администратора. Для этого необходимо входить как пользователь ADMIN:. Например, при необходимости зарегистрироваться на локальной машине, необходимо указать ADMIN:localhost. Если такая возможность отключена, администратор получит сообщение об ошибке.

Для включения новой возможности Database Mail нужно обратиться к странице Database Mail. Она позволяет отправлять электронную почту с вашего сервера баз данных с использованием SMTP-протокола. Хранимые процедуры, которые использует эта возможность, по умолчанию не включены и включаются именно здесь. (Поговорим подробнее об этой возможности в главе 5 “Автоматизация работы SQL Server”.)

На странице Native XML Web Services с помощью одной консоли можно быстро просмотреть активизированные Web-службы. Это дает возможность увидеть ненужные вам службы и отключить их. Страница Services Broker работает во многом аналогично очередям Service Broker.

Двумя самыми “опасными” хранимыми процедурами в SQL Server являются хранимые процедуры OLE Automation (те, имена которых начинаются с SP_OA и xp_cmdshell). Расширенные хранимые процедуры OLE Automation позволяют программисту T-SQL обрабатывать объекты внутри сценария T-SQL как объекты электронной почты и объекты третьей стороны. Вся эта функциональность теперь замещена CLR-интеграцией, но для сохранения обратной совместимости эта возможность осталась, хотя и в отключенном виде. Хранимая процедура xp_cmdshell используется хакерами и DBA для выхода на уровень DOS и запуска оттуда программ. Проблема заключается в том, что этим можно воспользоваться и для запуска команд, опасных для вашей машины.

Две другие унаследованные возможности также отключены по умолчанию. Это возможность SQL Mail, которая в SQL Server 2000 использовалась для отправки

почты с использованием MAPI-протокола, которая была заменена почтой Database Mail. И хранимые процедуры Web Assistant, позволяющие создать Web-страницу из SQL-запроса повсеместно, которые DBA не используются.

Для того чтобы активизировать большую часть установок, с которыми мы здесь познакомились, потребуется остановка и перезапуск настраиваемого экземпляра SQL Server.

Параметры запуска

SQL Server имеет набор ключей, которыми можно пользоваться при решении проблем или активизации дополнительных установок механизма базы данных. Ими можно воспользоваться при установке параметров работы службы или при запуске SQL Server из командной строки. Одним из способов изменить значение ключа является задание его в параметрах запуска SQL Server, что повлечет активизацию ключа при каждом входе в диспетчер SQL Server Configuration Manager. Для этого нужно дважды щелкнуть в SQL Server на странице SQL Server 2005 Services (по умолчанию MSSQLServer, но оно может меняться в зависимости от имени экземпляра) и перейти на вкладку Advanced. Необходимые ключи можно задать в параметре Startup Parameters, разделив их предварительно точками с запятыми. На рис. 4.4 видно, что один дополнительный ключ активизирован — это ключ: `-f`. Скоро мы займемся этим ключом детальнее.

Второй способ запуска SQL Server заключается в запуске программы `sqlservr.exe` непосредственно из командной строки. Этот файл по умолчанию располагается в папке `C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn`. Для него можно задать любой параметр, указав соответствующий ключ в командной строке после `sqlservr.exe`, как видно на рис. 4.5. Этот метод является наиболее предпочтительным методом запуска SQL Server в отладочном режиме, так как не останется незадействованным ни один из необходимых ключей. Остановить SQL Server можно с помощью комбинации клавиш `<Ctrl+C>` или закрыв окно с командной строкой. Никогда не запускайте SQL Server таким образом в эксплуатационном режиме, при завершении сеанса работы пользователя с машиной окно командной строки автоматически закроется, и SQL Server завершит свою работу.

Параметры, заданные при запуске SQL Server, действительно могут помочь в разрешении проблем. В этом разделе не ставится задача обсудить все ключи, но можно поговорить о наиболее используемых ключах. Остальные ключи можно узнать с помощью ключа `-?`. Например, с помощью ключей `-d` и `-l` можно поменять используемую SQL Server базу данных `master`.

```
SQLServr.exe -d C:\temp\TempMasterDB.mdf -lC:\temp\TempMasterLog.ldf
```

Ключ `-d` задает файл базы данных, ключ `-l` — имя файла регистрационного журнала. Это может потребоваться при использовании временной конфигурации базы данных `master`, если основная была разрушена. Другим полезным ключом является ключ `-T`, позволяющий активизировать заданные флаги трассировки для всех соединений экземпляра SQL. Им можно воспользоваться, например, для отслеживания взаимоблокировок в вашем экземпляре SQL Server (заметим, что этот флаг задается в верхнем регистре).

```
SQLServr.exe -T1204
```


Подробнее о флагах трассировки поговорим дальше в этой главе. Ключ `-g` используется для резервирования дополнительной памяти вне главного пула памяти SQL Server для использования, прежде всего расширенными хранимыми процедурами. Его не следует активизировать за исключением тех случаев, когда в журнале регистрации ошибок SQL Server такие сообщения не появлялись.

```
WARNING: Failed to reserve < n > bytes of contiguous memory.
WARNING: Failed to reserve contiguous memory of Size= 65536.
WARNING: Due to low virtual memory, special reserved memory
used < X > times since startup. Increase virtual memory on server.
```

Ключ `-m` переводит SQL Server в однопользовательский режим и задерживает процесс СНЕСКРОИПТ, который записывает данные с диска на устройство базы данных. Этот ключ можно использовать при восстановлении базы данных master из резервной копии. Ключ `-f` переводит SQL Server в минимальный режим, при котором разрешено только одно подключение. При переходе в минимальный режим SQL Server задерживает процесс СНЕСКРОИПТ и запрещает удаленные подключения. Вероятно, самой важной возможностью, отключающейся в минимальном режиме, являются пользовательские хранимые процедуры запуска. Администратор может обнаружить такую хранимую процедуру запуска, вызывающую проблему, не позволяющую полностью запускать SQL Server. Для этого можно перевести SQL Server в работу в минимальном режиме, удалить или исправить хранимую процедуру запуска, а затем запустить SQL Server снова уже без ключа, необходимого только для устранения возникшей проблемы.

Обязательно убедитесь в том, что до перевода SQL Server в однопользовательский режим SQL Server Agent был остановлен. В противном случае SQL Server Agent займет единственное имеющееся соединение.

Хранимые процедуры запуска

Хранимые процедуры запуска выполняют код T-SQL при запуске экземпляра SQL Server. Например, есть хранимая процедура запуска, отсылающая почту в момент запуска экземпляра. Пусковую хранимую процедуру можно использовать также для создания объектов в базе данных tempdb во время запуска SQL Server. Эти хранимые процедуры работают под управлением учетной записи, выполняющей роль sysadmin, и только sysadmin может создавать хранимые процедуры запуска. Сообщения об ошибках, генерируемых хранимыми процедурами, записываются в журнал регистрации ошибок SQL Server. Проверьте все здесь приведенные примеры на отладочном сервере и только потом на сервере, который находится в промышленной эксплуатации.

По умолчанию SQL Server не сканирует хранимые процедуры запуска, и для того, чтобы он это делал, необходимо запустить сценарий `sp_configure` следующим образом.

```
sp_configure 'scan for startup procs', 1
RECONFIGURE
```

После этого необходимо перезапустить экземпляр SQL Server, для того чтобы новая установка вступила в действие. Посмотрим, как это делается на простейшем примере. Сначала создадим таблицу `SQLStartupLog` в базе данных master, в которую записывается статистика во время каждого запуска SQL Server.

```
CREATE TABLE master.dbo.SQLStartupLog
(StartTime datetime)
GO
```

Затем создадим хранимую процедуру записи результатов работы хранимой процедуры SQLStartupLog в таблицу. Следующая хранимая процедура проделает это, сохраняя текущую дату в таблицу.

```
CREATE PROC master.dbo.InsertSQLStartupLog
as
INSERT INTO master.dbo.SQLStartupLog
SELECT GETDATE ()
```

И, наконец, последнее, что нужно сделать для использования хранимой процедуры sp_procoption: необходимо ее сделать хранимой процедурой запуска. Хранимая процедура sp_procoption использует только один параметр. Сначала необходимо определить, какая хранимая процедура будет задаваться, единственным задаваемым параметром будет параметр startup со значением 1 (активизировать) или 0 (отключено). Перед запуском следующих хранимых процедур задайте режим сканирования вашим SQL Server пусковых хранимых процедур.

```
sp_procoption @ProcName = 'master.dbo.InsertSQLStartupLog',
@OptionName= 'startup',
@OptionValue = 1
```

Затем перезапустите свой экземпляр SQL Server и сделайте запрос к master.dbo.SQLStartupLog для того, чтобы убедиться в том, что такая запись была сделана. Перед тем как закончить ознакомление с этим разделом, убедитесь в том, что эта установка отключена. Это можно сделать с помощью следующего запроса.

```
sp_procoption @ProcName = 'master.dbo.InsertSQLStartupLog',
@OptionName= 'startup',
OptionValue = 0
USE MASTER
GO
DROP TABLE master.dbo.SQLStartupLog
DROP PROC dbo.InsertSQLStartupLog sp_procoption
```

Восстановление системных баз данных

Время восстановления системных баз данных наступает при разрушении одной из них и отсутствии актуальных резервных копий. По существу, это позволит переустановить системные базы данных и избавиться от всего, что может привести к непредсказуемому поведению базы данных. В противном случае потребуются установить вновь все пакеты обновления, все пользовательские базы данных, включая базу данных поддержки служб Reporting Services. Кроме того, потребуются восстановить все учетные записи и настройки сервера.

Решение о восстановлении ваших системных баз данных нельзя отнести к категории простых решений. Это ответственное техническое решение, когда другие решения уже не дают никакого результата. При восстановлении системных баз данных они могут исчезнуть, но физически их файлы останутся в операционной системе и могут быть повторно подключены или восстановлены. Повторное включение баз данных является рутинным действием.

Для восстановления своих системных баз данных определите свою среду инсталляции SQL Server и перейдите в режим ввода командной строки. Из командной строки запустите setup.exe, как это делается при установке SQL Server, но при этом ей потребуется задать несколько новых ключей.

```
start /wait setup.exe /qn INSTANCENAME=<имя_экземпляра>
REINSTALL=SQL_Engine
REBUILDDATABASE=1 SAPWD=<новый_защищенный_пароль>
```

Ключ `/qn` подавляет вывод сообщений об ошибках и информационных сообщений, переадресуя их журналу регистрации ошибок. До полного восстановления базы данных на протяжении нескольких минут вы будете видеть пустой экран. Для просмотра сообщений во время выполнения операции восстановления необходимо добавить ключ `/qb`. При восстановлении системных баз данных для экземпляра по умолчанию в качестве имени экземпляра необходимо указать `MSSQLSERVER`.

После завершения восстановления баз данных необходимо вернуться к настройкам и базам данных, принятым по умолчанию. Необходимо восстановить базу данных `master` (подробнее об этом — в главе 18 “Резервное копирование и восстановление”) или повторно подключить указанную базу данных и восстановить учетные записи. Но лучше всего восстановить базу данных `master`. В таком случае ваши учетные записи и базы данных появятся автоматически.

Среда управления SQL Server Management Studio

Как вы, вероятно, уже знаете, именно со средой SQL Server Management Studio DBA проводит большую часть своего рабочего времени. Работая с этой средой, можно выполнить большую часть задач управления и проводить запуск запросов. Это логическое развитие среды Enterprise Manager (интерфейс управления SQL Server 2000), использующее облегченную версию Visual Studio 2005. Так как данная книга, прежде всего, ориентирована на профессионалов, все аспекты использования среды Management Studio рассматриваться не будут, но мы обязательно рассмотрим некоторые наиболее важные и новые возможности, которые можно использовать при администрировании SQL Server.

Отчеты

Одним из самых впечатляющих усовершенствований среды управления SQL Server являются интегрированные отчеты, являющиеся большим подспорьем DBA при выполнении задач области администрирования. Существует два уровня отчетности (серверный уровень и уровень баз данных), причем каждый из них работает как отчет служб Reporting Services внутри SQL Server Management Studio. Отчеты серверного уровня предоставляют информацию об экземпляре SQL Server и операционной системе. Отчеты уровня базы данных позволяют получить информацию о каждой базе данных в отдельности. При этом необходимо обладать доступом к любой базе данных, о которой вы хотите получить отчет, или ваша учетная запись должна иметь достаточно прав для запуска отчета серверного уровня.

Отчеты серверного уровня

Добраться до отчетов серверного уровня можно из вкладки Summary в Management Studio, нажав на левую клавишу мыши и установив курсор на экземпляре SQL Server, информацию о котором вы намереваетесь получить. Затем выберите элемент Report из раскрывающегося меню в окне Summary. Если окно Summary не активно, выберите элемент Summary в меню View.

Самым популярным отчетом на серверном уровне является отчет Server Dashboard, показанный на рис. 4.6. Отчет Server Dashboard предоставляет вам большое количество информации об экземпляре SQL Server 2005:

- какая версия SQL Server используется;
- все возможности этого экземпляра, которые не заданы SQL Server по умолчанию;

- ❑ статистика операций ввода/вывода и использования процессора по типу активности (произвольные запросы, работа со службами Reporting Services и т.д.);
- ❑ конфигурационная информация высокого уровня такого плана, как работа экземпляра кластера или использование им в своей работе динамического управления памятью.

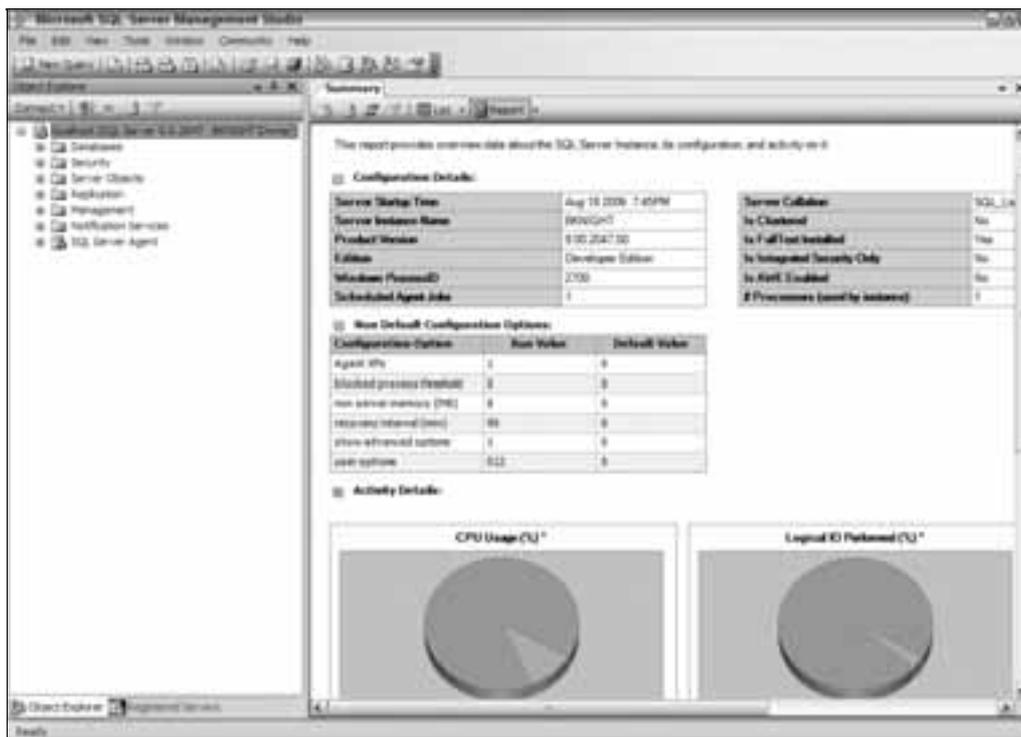


Рис. 4.6. Отчет Server Dashboard

Следует помнить о том, что отчеты серверного уровня и уровня баз данных, которые можно видеть из Management Studio, в некоторых случаях получают информацию из курсоров динамического управления. Вследствие этого, эти данные могут соответствовать последнему запуску SQL Server. Например, отчет Server Dashboard имеет несколько графиков, отражающих распределение загрузки процессора по типам запросов. Этот график не является историческим и отражает только использование процессора за период времени, когда SQL Server был включен.

Отчеты уровня баз данных

Отчеты уровня баз данных во многом аналогичны отчетам серверного уровня, но они выполняются тогда, когда база данных выбрана в Management Studio. С помощью этих отчетов можно увидеть информацию, соответствующую выбранной базе данных. Например, можно наблюдать все транзакции, которые выполняются в данный момент в базе данных, заблокированных пользователей или использование диска для данной базы данных, как показано на рис. 4.7.

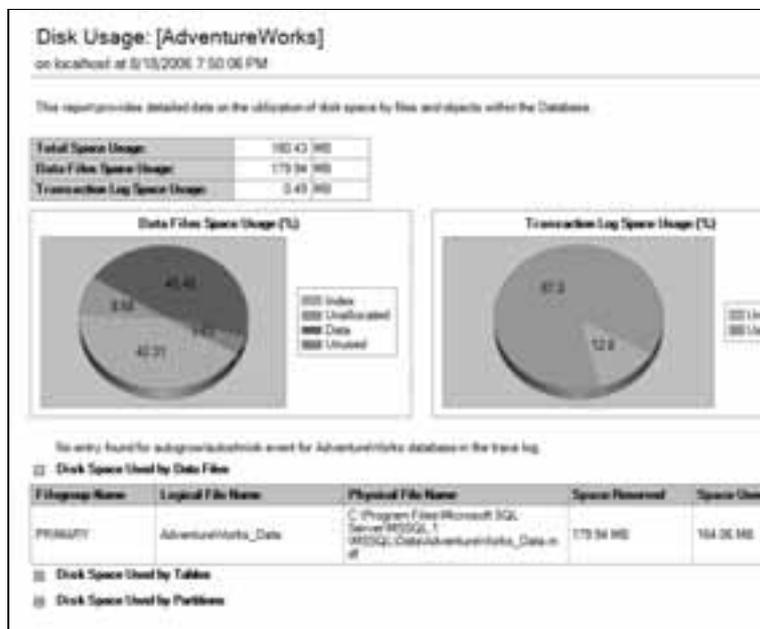


Рис. 4.7. Отчет об использовании дисков

Настройка SQL Server

Существует несколько способов настройки SQL Server. Об одном из таких механизмов уже было рассказано раньше. Это работа с диспетчером SQL Configuration Manager и инструментальным средством Surface Area Configuration. Эти две возможности позволяют активизировать различные возможности и службы. Однако они не позволяют настроить отдельные экземпляры. Для механизма базы данных существует два основных метода настройки экземпляра: использование хранимой процедуры `sp_configure` и работа с экраном Server Properties. Для того чтобы получить доступ к экрану Properties, находясь в Management Studio, необходимо нажать правую клавишу мыши на механизме базы данных и выбрать Properties. Но перед настройкой вашего экземпляра нужно быть предельно осторожным. При работе с некоторыми из этих настроек можно нарушить работу экземпляра или его безопасность. В этом разделе речь пойдет о нескольких более важных настройках, о которых речь будет идти на протяжении всей книги.

Работать с экраном Server Properties быть может удобней, но он не имеет всех возможностей хранимой процедуры `sp_configure`. Вкладка General на экране Server Properties предоставляет такую информацию об экземпляре SQL Server, как версия установленного SQL Server и наличие кластеризации вашего экземпляра, в которую нельзя внести изменения. Он также отображает такую информацию о сервере, как количество используемых процессоров и объем памяти компьютера. Следует помнить, что наличие на вашем сервере установленных 2 Гбайта ОЗУ никак не означает, что вся эта память находится в распоряжении SQL Server.

На странице Memory сценария Server Properties можно увидеть, сколько памяти может использовать SQL Server. По умолчанию SQL Server настроен на максимальное использование SQL Server имеющейся памяти. Обычно принимается решение об установке минимального объема памяти, которую используют экземпляры во

время старта. Этот экран также позволяет включать использование AWE. AWE позволяет наращивать используемую память при работе SQL Server на 32-разрядных машинах после преодоления ограничения на 4 Гбайта ОЗУ (Подробнее об этом в главе 11 “Оптимальная настройка сервера”)

На странице `Processors` имеется возможность использовать имеющиеся процессоры SQL Server для операций ввода/вывода или для операций с потоками. Такая возможность очень кстати при наличии восьми и более процессоров и более одного экземпляра SQL Server. Таким образом, можно задать использование четырех процессоров одним экземпляром, а остальных процессоров — другим экземпляром. В некоторых случаях при наличии большого количества параллельных подключений к вашему SQL Server можно задать параметр `Maximum Worker Threads`. Задание значения 0 (умолчание) позволяет SQL Server автоматически и динамически определять оптимальное количество потоков, обрабатываемых одним процессором. Эти потоки используются для обслуживания соединений и таких системных функций, какой, например, является выполнение команды `SHESKPOINT`. В общем случае, если это значение не трогать, вы получите оптимальную производительность. Кроме того, для того, чтобы Windows выделяла процессу SQL Server высший приоритет, можно также выбрать параметр `SQL Server Priority`. Эта установка может просто “напрашиваться” для изменения, но это следует делать только после самого тщательного тестирования.

Осуществить настройку метода аутентификации SQL Server можно на странице `Security`. Есть два варианта: аутентификация с применением аутентификации SQL Server и Windows или только аутентификация с применением аутентификации Windows. Этот же вопрос задается во время установки экземпляра, а этот экран предоставляет дополнительную возможность поменять установки. В разделе `Login Auditing` всегда будет, как минимум, выбрана возможность `Failed Logins Only`. Это позволяет SQL Server отслеживать ввод ошибочного пароля или попытки вторжения. (Подробнее об этом говорится в главе 9 “Безопасность механизма баз данных”.)

На странице `Connections` (рис. 4.8) есть возможность настройки стандартных свойств подключения. Одной из очень удобных настроек является довольно редко применяющаяся настройка `Query Governor`. Эта настройка может немного сбивать с толку DBA. Она сообщает SQL Server, что если выполнение какого-либо запроса SQL Server по оценке может занять больше определенного времени, он будет остановлен еще до выполнения. Эту установку можно настроить без перезапуска экземпляра или обойти на уровне соединения, если соединение использует следующий синтаксис и выполняется под управлением учетной записи `sysadmin`.

```
SET QUERY_GOVERNOR_COST_LIMIT 120
```

Этим запросом задается выполнение SQL Server только тех запросов, для выполнения которых потребуется менее 120 секунд. При отправке на SQL Server запроса, выполнение которого может потребовать больше этого времени (скажем, 200 секунд), пользовательский запрос выполнен не будет, и пользователь, отправивший такой запрос, получит следующее сообщение об ошибке.

```
Msg 8649, Level 17, State 1, Line 4
The query has been canceled because the estimated cost of this query
(200) exceeds
the configured threshold of 120. Contact the system administrator.
```

Так как данная установка учитывает только оцениваемые затраты запроса, она может оказаться неточной. Оптимизатор запросов может дать оценку, что для выполнения запроса потребуется 2 секунды, но в действительности для его выполнения в зависимости от системных условий иногда может потребоваться до 45 секунд. На

этом экране, конечно, можно установить параметры соединения по умолчанию, если установки соединения недостаточно точно определены. Лучше всего здесь задать значение `SET NOCOUNT`. При этой установке клиенту не будет отправляться сообщение “8 Rows Affected”, если оно не было запрошено специально. Это также повлечет незначительное улучшение производительности, так как для этого требуется отправка дополнительного набора записей, отправляемых с SQL Server, который к тому же может повлечь за собой нежелательный трафик.

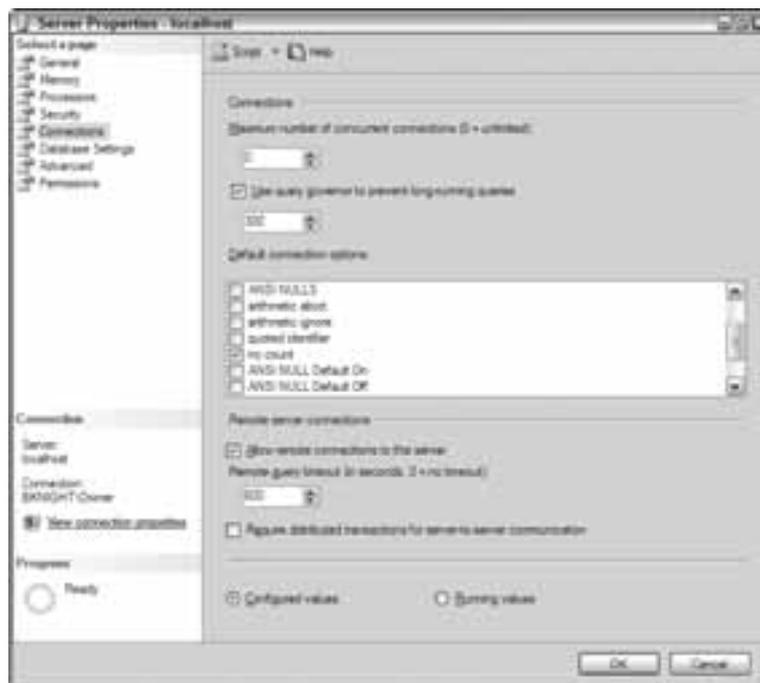


Рис. 4.8. Страница Connections

На странице Database Settings есть отличная возможность задания мест размещения баз данных по умолчанию и регистрационных журналов. Специалисты говорят о целесообразности хранения данных и регистрационных журналов на различных устройствах. Это позволит получить максимальный выигрыш при использовании RAID-систем, о которых речь пойдет в главе 11 “Оптимальная настройка сервера”. Однако по умолчанию файлы данных и регистрационных журналов хранятся на одном устройстве в папках `%System Drive%\Program Files\Microsoft SQL Server\MSSQL.1\Data` и `Log`.

При желании экземпляр можно настроить с помощью хранимых процедур `sp_configure`. Эту команду можно запускать без параметров. Таким образом, можно посмотреть текущие установки. При этом по умолчанию вы увидите основные настройки (около 14 для Service Pack 1). Многие настройки, которые вы хотите просмотреть, считаются дополнительными. Для того чтобы просмотреть их, необходимо настроить экземпляр таким образом, чтобы эти дополнительные параметры отображались. Это можно сделать с помощью следующей команды.

```
sp_configure 'show advanced options', 1
RECONFIGURE
```

Установка не перейдет из состояния настройки в рабочее состояние, пока не будет запущена команда RECONFIGURE. В некоторых случаях для установок, для активизации которых потребуется перезапуск экземпляра SQL Server, можно запустить команду RECONFIGURE WITH OVERRIDE. При попытке сделать одну из этих установок, будет выдано следующее сообщение.

```
Msg 5807, Level 16, State 1, Line 1
Recovery intervals above 60 minutes not recommended.
Use the RECONFIGURE WITH OVERRIDE statement to force this
configuration.
```

Вот программный код, который позволяет обойти эти установки. После этого, для того, чтобы активизировать эту установку, потребуется перезапустить ваш экземпляр SQL Server.

```
EXEC sp_configure 'recovery interval', 90
RECONFIGURE WITH OVERRIDE
GO
```

Фильтрация объектов

В SQL Server 2000 для поиска списка объектов в Enterprise Manager используются объектные модели DMO и SQL-NS. Когда насчитываются десятки или сотни отображаемых объектов, для отображения только их перечня в Enterprise Manager могут потребоваться минуты, так как старые объектные модели не были рассчитаны на отображение такого количества объектов. Это становится затруднительно при увеличении масштаба применения SQL Server, если SQL Server был адаптирован хост-провайдерами и большими предприятиями, ведущими сотни баз данных на одном сервере.

Среда SQL Server 2005 Management Studio использует в работе новую объектную модель, которая называется SQL Server Management Objects (SMO) и масштабируется значительно лучше предыдущих версий объектной модели. Например, при работе Management Studio можно расширить список таблиц и, ожидая расширения дерева, перейти к выполнению следующей задачи. Когда таблицы полностью готовы к обновлению, меню расширяется, а выполнение вашей задачи, работающей параллельно, при этом не прерывается.

Кроме того, упрощена операция фильтрации объектов, что очень удобно, когда в вашем распоряжении имеются десятки объектов. Для фильтрации объектов в Management Studio необходимо выбрать узел дерева, которое будет фильтроваться, и, щелкнув на пиктограмме Filter в Object Explorer, открыть диалоговое окно Object Explorer Filter Settings (рис. 4.9). Это позволит производить фильтрацию по имени, схеме или времени создания объекта. Раскрывающийся список позволяет выбрать объект фильтрации, после чего следует ввести соответствующее имя в столбце Value.

Журналы регистрации ошибок

Как вы, вероятно, уже испытали на собственном опыте, когда происходит какой-либо сбой в работе приложения, первой “жертвой” обычно становится база данных. DBA нужно доказывать, что это не его ошибка. Первое, что может сделать DBA, — это подключиться к серверу и просмотреть журнал регистрации ошибок экземпляра SQL Server, а затем журналы событий Windows.

В SQL Server 2005 в Management Studio вы имеете возможность быстрого просмотра журналов регистрации в консолидированном виде. Для просмотра этих журналов щелкните правой кнопкой мыши на SQL Server Logs в дереве Management и

выберите View⇒SQL Server and Windows Log. Откроется экран Log File Viewer, где можно пометить или снять отметку с файлов регистрационных журналов, которые будут просматриваться. Возможность объединения регистрационных журналов, полученных в SQL Server Agent, Database Mail и Windows Event Viewer, показана на рис. 4.10.

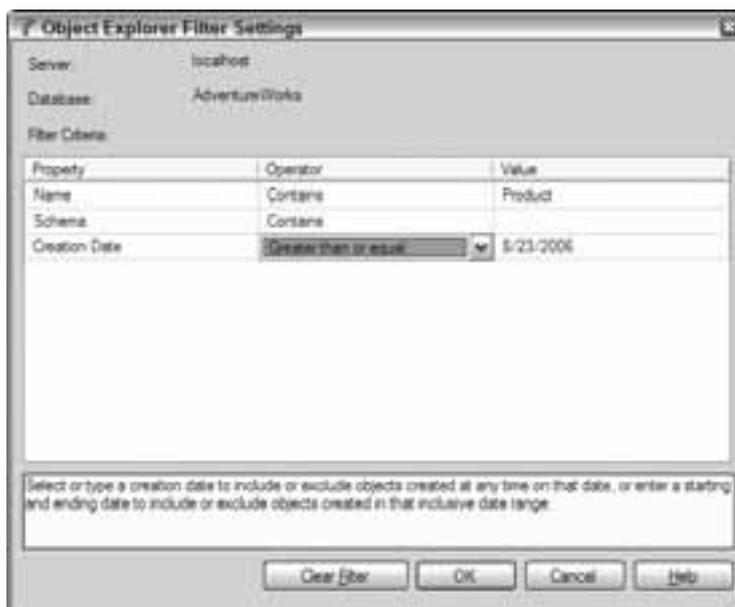


Рис. 4.9. Диалоговое окно Object Explorer Filter Settings

Есть ситуации, когда для определения причины сбоев в работе приложений требуется слить регистрационные журналы с нескольких машин на одну. Для этого можно щелкнуть на кнопке Load Log и просмотреть файл .LOG. Это может быть регистрационный журнал ошибок Windows, который был переведен в формат .LOG, или объединенный регистрационный журнал SQL различных экземпляров. Таким образом, например, можно объединить все имеющиеся на одном сервере журналы SQL для глобального просмотра всех физических проблем, имевших место на данной машине.

Инструментальное средство Activity Monitor

Инструментальное средство Activity Monitor позволяет производить просмотр соединений, активных на данный момент. Этот монитор может использоваться для обнаружения процесса, блокирующего работу других процессов. Для того чтобы открыть Activity Monitor в Management Studio, щелкните дважды на элементе Activity Monitor дерева Management.

Это инструментальное средство является отличным способом контроля за всеми пользователями, подключившимися к вашей машине, и за их деятельностью. На странице Process Info (показанной на рис. 4.11), можно наблюдать подключение через учетную запись (называется SPID) к вашей машине. В обилии информации, представленной в этом окне, можно даже запутаться. Для просмотра всей информации окно имеет возможность прокрутки слева направо. Долго и нецелесообразно объяснять значение каждого столбца, но на этапе отладки интерес может представлять любой из этих столбцов.

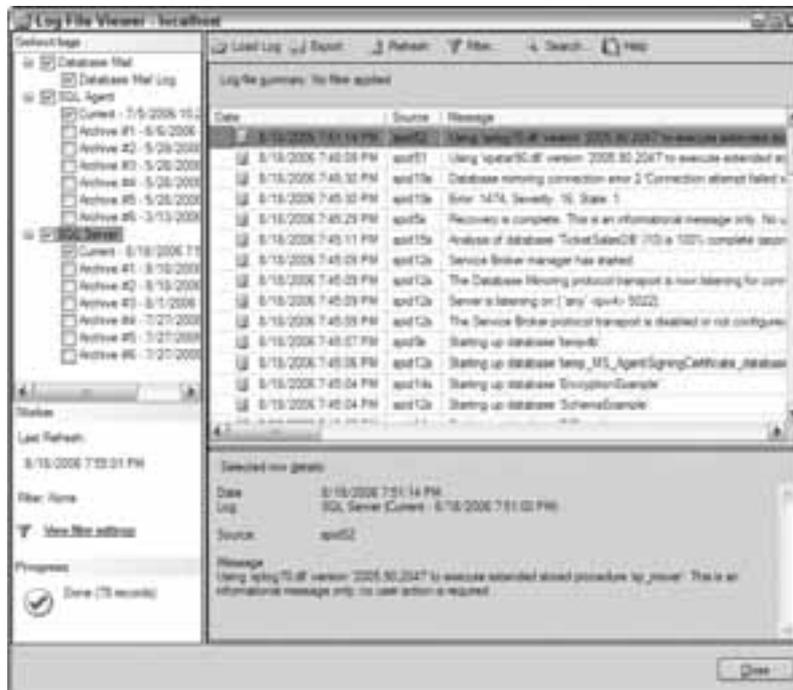


Рис. 4.10. Диалоговое окно Log File Viewer

- ❑ **Process ID:** уникальный номер, присвоенный процессу, подключенному к SQL Server. Он также называется SPID. Сразу же за номером должна следовать пиктограмма, отражающая состояние соединения. Изображение песочных часов является подсказкой о том, что этот процесс находится в состоянии ожидания или заблокирован другим соединением.
- ❑ **System Process:** отражает факт подключения других процессов, являющихся внутренними процессами SQL Server. По умолчанию эти процессы отфильтровываются щелчком на кнопке Filter и должны добавляться, когда требуется просмотреть внутренние установки SQL Server.
- ❑ **User:** имя пользователя в базе данных, к которой привязан процесс.
- ❑ **Database:** имя базы данных, к которой подключен пользователь.
- ❑ **Status:** отражение активности пользователя. Отсутствие активности пользователя (sleeping) означает, что SQL Server сохраняет это соединение активным, несмотря на то, что никакой активности от данного пользователя не исходит. Активность в свою очередь означает, что данный пользователь в данный момент выполняет какой-то запрос. Есть и другие состояния, но о них речь пойдет в этой главе позднее, когда будем говорить о настройке рабочих характеристик системы.
- ❑ **Open Transactions:** показывает количество транзакций, открытых в данной точке пользователем.
- ❑ **Command:** отображает тип команды, выполняемой в данный момент. Так можно увидеть команды и операторы SELECT, DBCC, INSERT или AWAITING COMMAND. Таким образом нельзя увидеть реальный запрос, выполняющийся

пользователем, но можно судить о типе активности, которая выполняется на вашем сервере.

- ❑ **Application:** отражает наименование приложения, которое использует пользователь для подключения к вашему экземпляру. Оно задается разработчиком в строке подключения, которым пользуется это приложение.
- ❑ **Wait Time:** если процесс заблокирован или ожидает завершения другого процесса, этот столбец отражает время ожидания данного процесса. Тип ожидания отражает тип ожидаемого события.
- ❑ **CPU:** объем процессорных ресурсов, затраченных на этот процесс с момента регистрации.
- ❑ **Login Time:** время запуска процесса.
- ❑ **Last Batch:** время выполнения процессом последнего запроса.
- ❑ **Physical I/O:** объем операций ввода/вывода, произведенных данным процессом с момента регистрации.
- ❑ **Memory Usage:** объем памяти, задействованной процессом с момента регистрации.
- ❑ **Host:** имя рабочей станции, с которой было произведено соединение, или имя сервера. Это действительно полезное значение, но в некоторых случаях, когда к вашему SQL Server подключается Web-сервер, оно теряет свою актуальность.
- ❑ **Blocked By:** показывает идентификатор процесса (SPID), заблокировавшего это соединение.
- ❑ **Blocking:** отражает количество процессов, заблокированных этим соединением.

Для обновления данных можно щелкнуть на кнопке Refresh. Для автоматического обновления Activity Monitor через каждые 60 секунд (или какой-либо другой заданный интервал времени) нужно щелкнуть на ссылке View Refresh Settings. Но при этом существует опасность переусердствовать: поэтому не задавайте слишком малый интервал обновления (как, например, через каждые две секунды), так как это повлечет за собой задержку в работе других активных систем из-за частого повторения запроса. Для отражения информации об определенных узлах, подключенных учетных записях или подключениях можно воспользоваться фильтрами, задавая там нижний предел используемых ими ресурсов. И, наконец, информацию можно отсортировать, щелкнув на определенном столбце.

Двойной щелчок на любом процессе позволит просмотреть последний запрос, который запускался данным соединением. Кроме того, этот экран позволяет щелкнуть на кнопке Kill Process для отключения пользователя. Процессы также отключаются щелчком правой клавиши по пользователю или щелчком на кнопке Kill Process с главной страницы Activity Monitor. После чего пользователь, запустивший запрос, увидит на своем экране следующее сообщение.

```
Msg 0, Level 11, State 0, Line 0
A severe error occurred on the current command. The results, if any,
should be
discarded.
```

Страницы Locks by Process и Locks by Object позволяют просмотреть заблокированные таблицы, сгруппированные по идентификатору SPID или по имени объекта. Это окно позволяет выбрать заблокированную таблицу, а затем отобразить детальную информацию о блокировках на этом объекте. После этого для того, чтобы прервать соединение, вызвавшее эту блокировку, можно вернуться к исходному экрану.

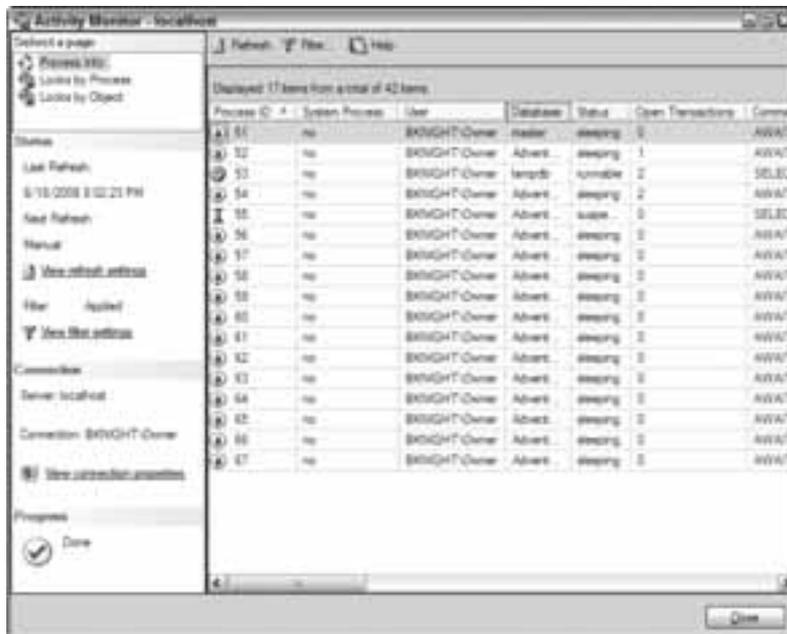


Рис. 4.11. Диалоговое окно Activity Monitor. Просмотр блокировок

Для того чтобы узнать, как работает этот экран, можно смоделировать возникновение блокировки запуском следующего запроса из окна запросов при подключении к базе данных AdventureWorks (после выполнения этих шагов обязательно восстановите исходное состояние базы данных AdventureWorks).

```
BEGIN TRAN
DELETE from Production.ProductCostHistory
```

Этот запрос возвратит количество обработанных записей. Заметим также, что этот запрос специально делается таким образом, что он не будет отработан. Другими словами, есть оператор `BEGIN TRAN`, но отсутствует `ROLLBACK` или `COMMIT`. Без этих операторов таблица блокируется. Обычно запросы запускаются без оператора `BEGIN TRAN`. В таких ситуациях имеет место неявная транзакция, которая выполняется сразу же после завершения пакета операторов.

Затем, не закрывая первого окна, откройте другое окно запросов и запустите следующий запрос к базе данных AdventureWorks.

```
SELECT *
FROM Production.ProductCostHistory
```

Этот запрос никогда не возвратит результата и зависнет. Не закрывайте эти окна. Заметим, что вверху каждого окна отображается то, под какой учетной записью вы зарегистрировались, а идентификатор вашего процесса выводится в скобках. При открытых окнах этих запросов перейдите к диалоговому окну Activity Monitor и посмотрите, какой вид имеют это подключения.

Откройте диалоговое окно Activity Monitor и обратите внимание на отметку в виде песочных часов рядом с одним из соединений. Щелкните правой кнопкой мыши по этому соединению и выберите в появившемся меню `Kill Process`. В окне запросов, лежащем под Activity Monitor, появится вышеописанная ошибка. Затем перейдите на страницу `Locks by Object` и выберите из раскрывающегося меню объектов таблицу

Production.ProductCostHistory. Таким образом, можно увидеть все основные блокировки, открытые в данный момент для этой таблицы.

Заметим, что в столбце Mode появится X, как видно на рис. 4.12. Значок X означает, что этот ключ имеет исключительную блокировку (в данном случае имеется сотня таких блокировок). Исключительная блокировка также означает, что никто не будет иметь права на чтение этих данных. При другом режиме, обозначенном как S, SQL Server представляет общую блокировку, которая в большинстве случаев совершенно безобидна, при этом все пользователи имеют право просмотра одних и тех же данных. Пользователь может запросить “грязное” чтение (dirty read) данных незавершенных транзакций. Этого можно добиться добавлением предложения WITH (NOLOCK).

```
SELECT *
FROM Production.ProductCostHistory
WITH (NOLOCK)
```

Столбец Process ID позволяет узнать идентификатор процесса, вызвавшего проблему. Вы можете “взять” этот идентификатор и, перейдя к странице Process Info, удалить это соединение. Столбец Type позволяет узнать, какие процессы находятся в данный момент в заблокированном состоянии. SQL Server запускается на нижнем уровне блокировки, который является блокировкой KEY, а затем переходит к более высоким уровням блокировки EXTENT, TABLE и, наконец, DATABASE.

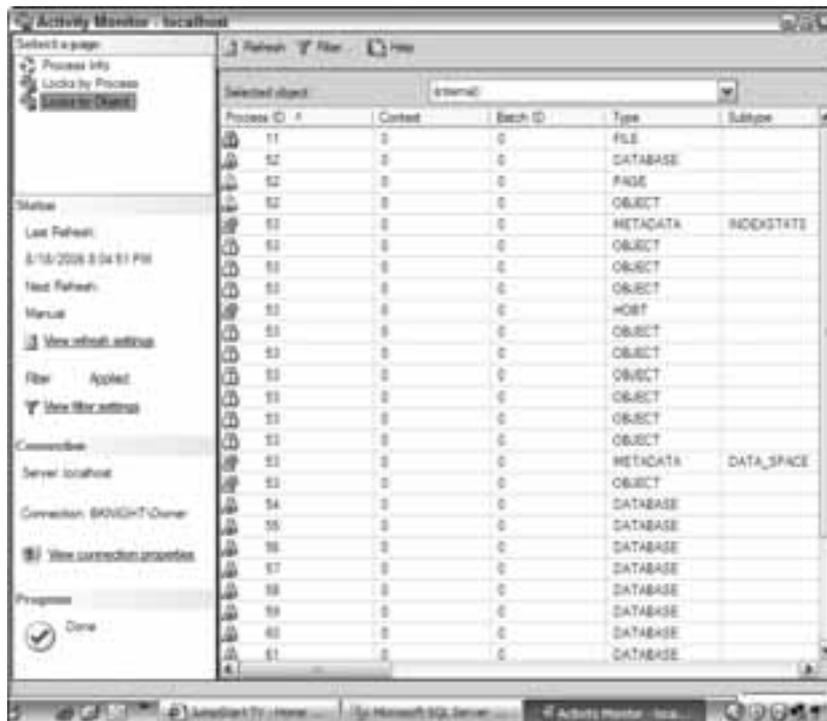


Рис. 4.12. Просмотр блокировок в диалоговом окне Activity Monitor

Управление процессами в T-SQL

Управлять работой сервера можно и с помощью языка T-SQL. Зачастую DBA предпочитают именно этот способ в качестве быстродействующего средства лечения “долгоиграющих” запросов или пользователей, жалующихся на плохую производительность. Причина этого заключается в том, что он предоставляет значительно больше гибкости, чем средство Activity Monitor.

Хранимые процедуры `sp_who` и `sp_who2`

Хранимая процедура `sp_who` аналогично Activity Monitor позволяет посмотреть, кто подключился к вашему экземпляру. Но для этого можно воспользоваться недокументированной хранимой процедурой `sp_who2`, которая представляет значительно более полную информацию о каждом процессе. Независимо от используемой процедуры пользуйтесь одними и теми же параметрами. Памятуя при этом о том, что хранимая процедура `sp_who` отображает только часть информации, воспользуемся хранимой процедурой `sp_who2`.

Для того чтобы увидеть все подключения к серверу, запустим хранимую процедуру `sp_who2` без каких-либо параметров. Это позволяет просмотреть данные в том же ракурсе, что и Activity Monitor. Для просмотра только активных подключений к серверу задается параметр 'active'.

```
sp_who2 'active'
```

Наконец, для просмотра детальной информации о процессе можно задать идентификатор процесса.

```
sp_who2 55
```

Курсор `sys.dm_exec_connections`

Курсор динамического управления `sys.dm_exec_connections` позволяет получить еще больше информации, которая может быть использована для устранения проблем механизма баз данных SQL Server. Этот курсор возвращает по одной строке на один сеанс в SQL Server. Так как это курсор, он делает вывод в табличном виде, позволяя создавать запросы с фильтрацией нужных данных, как показано в этом запросе, выводящий информацию только о тех пользовательских подключениях, которые выполняли операцию записи.

```
SELECT * FROM  
sys.dm_exec_sessions WHERE is_user_process = 1  
AND writes > 0
```

Кроме той информации, которую можно просмотреть с использованием всех прежде упомянутых способов просмотра процессов, этот курсор позволяет просмотреть количество строк, выбранных пользователем после открытия соединения, количество операций чтения, записи и логического чтения. Этот курсор также позволяет просмотреть установки, сделанные для каждого соединения, и какова была последняя ошибка.

Команда `DBCC INPUTBUFFER`

Команда `DBCC INPUT BUFFER` — это команда, позволяющая просмотреть, какую команду SQL выполняет определенный процесс. Эта команда принимает только один входной параметр. Как показано в следующем запросе, это идентификатор диагностируемого процесса.

DBCC INPUTBUFFER (53)

Самым важным столбцом, который возвращает эта команда, является столбец EventInfo, содержащий описание события. При запуске обычного пакета большого размера по умолчанию в окне запросов вы увидите только 256 первых символов.

```

EventInfo Parameters EventInfo
-----
-----
Language Event 0 begin tran delete from production.productcosthistory
(1 row(s) affected)

```

Функция динамического управления `sys.dm_exec_sql_text`

Иногда результаты работы DBCC INPUTBUFFER могут потребоваться в табличном виде. Для этих целей можно воспользоваться функцией динамического управления `sys.dm_exec_sql_text`. Она позволяет найти текст определенного запроса. Ее можно использовать совместно с курсором динамического управления `sys.dm_exec_query_stats` для выборки самых медленных запросов в базе данных. Запрос, изображенный внизу (этот запрос можно найти на странице, посвященной этой книге на Web-узле www.wrox.com), производит выборку количества отработок запроса, среднее время использования процессора и длительность запроса. Вот текст этого запроса.

```

SELECT TOP 10 execution_count [Number of Executions],
total_worker_time/execution_count AS [Average CPU Time],
Total_Elapsed_Time/execution_count as [Average Elapsed Time],
(SELECT SUBSTRING(text,statement_start_offset/2,
(CASE WHEN statement_end_offset = -1 then
LEN(CONVERT(nvarchar(max), text)) * 2
ELSE statement_end_offset
end -statement_start_offset)/2)
FROM sys.dm_exec_sql_text(sql_handle)) AS query_text
FROM sys.dm_exec_query_stats
ORDER BY [Average CPU Time] DESC

```

Курсор динамического управления `sys.dm_exec_query_stats` также отображает информацию о нагрузке и другую полезную информацию. С его помощью можно просмотреть план выполнения каждого выполненного запроса. В функции `sys.dm_exec_sql_text` можно воспользоваться столбцом `sql_handle` из этого курсора. Так как этот запрос работает на уровне планирования, когда кто-либо делает какие-либо изменения в тексте запроса, этот запрос отображается новой строкой уже как новый запрос.

Флаги трассировки

Флаги трассировки предоставляют собой дополнительные механизмы доступа к скрытым возможностям SQL Server и реализации тактики устранения неисправностей. В некоторых случаях они позволяют изменить рекомендуемое поведение SQL Server и включить такие возможности, как поддержка сетевых устройств для файлов баз данных. В других случаях флаги трассировки используются для включения дополнительного управления, так как некоторые флаги позволяют диагностировать взаимоблокировки. Для активизации флага трассировки можно воспользоваться командой DBCC TRACEON с последующим указанием нужного флага.

DBCC TRACEON (1807)

Для деактивизации флага используется команда `DBCC TRACEOFF`. После нее задаются деактивируемые флаги трассировки (список флагов, разделенных запятыми).

```
DBCC TRACEOFF (1807, 3604)
```

Активируя трассировку по умолчанию, вы активизируете ее для одного соединения. Например, при активизации флага трассировки 1807, позволяющего диагностировать взаимоблокировки, вы сможете только диагностировать взаимоблокировки в рамках соединения, для которого была дана команда `DBCC TRACEON`. Трассировку можно активизировать также на уровне всего сервера. Для этого необходимо задать дополнительный ключ `-1`.

```
DBCC TRACEON (1807, -1)
```

После активизации трассировки следующим шагом должна быть проверка работы трассировки. Для этого можно воспользоваться командой `DBCC TRACESTATUS`.

```
DBCC TRACESTATUS (3604)
```

В случае если трассировка не работает, эта команда возвратит следующий результат.

```
TraceFlag Status Global Session
-----
3604 0 0 0
(1 row(s) affected)
```

При необходимости проверить все трассировки, относящиеся к этому подключению, задайте эту команду с параметром `-1`, как показано ниже.

```
DBCC TRACESTATUS (-1)
```

Как видно из следующих результатов работы этого запроса, две трассировки включены. Флаг трассировки 1807 активизирован глобально для всех подключений к SQL Server, а флаг трассировки 3604 активизирован только для текущего сеанса.

```
TraceFlag Status Global Session
-----
1807 1 1 0
3604 1 0 1
(2 row(s) affected)
```

Если трассировки отключены, вы получите следующее сообщение.

```
DBCC execution completed. If DBCC printed error messages, contact your
system administrator.
```

Но если у вас нет для этого специального разрешения от компании Microsoft, ваш экземпляр SQL Server не сможет неограниченно активизировать флаги трассировки. Это может привести к непредсказуемому поведению сервера, если эти флаги активизированы постоянно. Кроме того, флаг, который вы используете в своей работе сейчас в следующих версиях или обновлениях SQL Server тоже может быть отключен. Находясь в отладочном режиме, вы можете активизировать любой флаг трассировки из командной строки в момент запуска SQL Server. Как уже говорилось в этой главе, SQL Server можно запустить в режиме трассировки с помощью команды `sqlservr.exe` из командной строки, задав после нее ключ `-T`.

К настоящему моменту упоминалось несколько флагов трассировки, мы не останавливались на деталях их использования. По мере изучения этой книги мы по-

знакомим вас с практическим применением других флагов трассировки. Но самыми популярными флагами, без сомнения, будут флаги, связанные с трассировкой взаимных блокировок.

Взаимная блокировка происходит тогда, когда один и более ресурсов пытаются получить доступ к элементу, который постоянно заблокирован другим ресурсом. Тогда соответствующее подключение может быть отключено, а клиент выдаст сообщение о взаимоблокировке. Подробнее о том, что из себя представляют взаимоблокировки и как они могут повлиять на работоспособность вашего экземпляра, будет рассказано в главах 6 “Администрирование служб Integration Services и настройка производительности” и 7 “Администрирование и настройка служб Analysis Services”, но на этом этапе следует запомнить, что активизация флага трассировки 1204 вызывает взаимоблокировку, вследствие чего в журнал регистрации ошибок SQL Server будет отправлено сообщение, аналогичное следующему.

```
Deadlock encountered .... Printing deadlock information
Wait-for graph
Node:1
RID: 8:1:140:0 CleanCnt:3 Mode:X Flags: 0x2
Grant List 1:
Owner:0x03364B20 Mode: X Flg:0x0 Ref:0 Life:02000000 SPID:59 ECID:0
XactLockInfo: 0x059D67D4
SPID: 59 ECID: 0 Statement Type: UPDATE Line #: 10
Input Buf: Language Event: exec p2
Requested By:
ResType:LockOwner Stype: 'OR'Xdes:0x059D6248 Mode: U SPID:57 BatchID:0
ECID:0 TaskProxy:(0x04AEE33C) Value:0x335cfe0 Cost:(0/216)
Node:2
RID: 8:1:138:0 CleanCnt:2 Mode:X Flags: 0x2
Grant List 1:
Owner:0x03364620 Mode: X Flg:0x0 Ref:0 Life:02000000 SPID:57 ECID:0
XactLockInfo: 0x059D626C
SPID: 57 ECID: 0 Statement Type: UPDATE Line #: 10
Input Buf: Language Event: exec p1
Requested By:
ResType:LockOwner Stype: 'OR'Xdes:0x059D67B0 Mode: U SPID:59 BatchID:0
ECID:0 TaskProxy:(0x05FD433C) Value:0x33645e0 Cost:(0/116)
Victim Resource Owner:
ResType:LockOwner Stype: 'OR'Xdes:0x059D67B0 Mode: U SPID:59 BatchID:0
ECID:0
TaskProxy:(0x05FD433C) Value:0x33645e0 Cost:(0/116)
```

Это сообщение свидетельствует о том, что у вас имеются серьезные проблемы с запросами или группой запросов. Оно содержит SPID-идентификатор процесса (упоминавшийся ранее как идентификатор процессов в предыдущих разделах), заблокировавшего другие процессы (которые называются жертвой). В этом случае жертвой является процесс, имеющий SPID-идентификатор, равный 59, результаты этого запроса утеряны, поэтому этот запрос требуется запустить повторно.

Флаг трассировки 1222 возвращает в журнал регистрации ошибок больше информации, чем флаг 1204, формируя его в формате XML-документа (хотя он и не соответствует XSD-описанию). Результаты очень полные, и именно поэтому обычно этот флаг активизируется только после того, как данных полученных с помощью флага 1204 для решения возникшей проблемы, оказалось недостаточно. Результаты, полученные с помощью этого флага, помогут разобраться, какие запросы стали жертвами, а какие — мешали им работать и соответствуют такому фрагменту журнала регистрации ошибок.

```
deadlock-list
deadlock victim=process279b098
process-list
process id=process279b098 priority=0 logused=116 waitresource=RID:
7:1:141:0 waittime=10000 ownerId=22637
transactionname=user_transaction
lasttranstarted=2005-01-25T16:39:22.050 XDES=0x529e7e0 lockMode=U
schedulerid=2
kpid=2120 status=suspended spid=55 sbid=0 ecid=0 transcount=2
lastbatchstarted=2005-01-25T16:39:35.357 lastbatchcompleted=2005-01-
25T16:36:55.217
clientapp=SQL Server Management Studio - Query hostname=USER13
hostpid=2852
loginname=DOMAIN\user1 isolationlevel=read committed (2)
xactid=236223201313
currentdb=7 lockTimeout=4294967295 clientoption1=671090784
clientoption2=390200
executionStack
frame procname=general.dbo.p2 line=10 stmtstart=186 stmtend=228
sqlhandle=0x03000700e8ea0d065c0a1101e79500000100000000000000
UPDATE table1 SET column1=5
frame procname=ad hoc line=1
sqlhandle=0x0100070068747519405a600300000000000000000000000000
exec p2
inputbuf
exec p2
process id=process279b168 priority=0 logused=216 waitresource=RID:
7:1:144:0 waittime=2453 ownerId=22635 transactionname=user_transaction
lasttranstarted=2005-01-25T16:39:19.190 XDES=0x529e258 lockMode=U
schedulerid=2
kpid=164 status=suspended spid=53 sbid=0 ecid=0 transcount=2
lastbatchstarted=2005-
01-25T16:39:32.503 lastbatchcompleted=2005-01-25T16:39:08.627
clientapp=SQL Server
Management Studio - Query hostname=USER13 hostpid=3192
loginname=DOMAIN\user1
isolationlevel=read committed (2) xactid=227633266690 currentdb=7
lockTimeout=4294967295 clientoption1=671090784 clientoption2=390200
executionStack
frame procname=general.dbo.p1 line=10 stmtstart=182 stmtend=224
sqlhandle=0x03000700afc619055c0a1101e79500000100000000000000
UPDATE table1 SET column1=2
frame procname=ad hoc line=2 stmtstart=6
sqlhandle=0x01000700d6e53430205b600300000000000000000000000000
exec p1
inputbuf
exec p1
resource-list
ridlock fileid=1 pageid=144 dbid=7 objectname=general.dbo.t2
id=lock368ab80
mode=X associatedObjectId=72057594038517760
owner-list
owner id=process279b098 mode=X
waiter-list
waiter id=process279b168 mode=U requestType=wait
ridlock fileid=1 pageid=141 dbid=7 objectname=general.dbo.t1
id=lock368ac80
mode=X associatedObjectId=72057594038452224
```

```
owner-list
owner id=process279b168 mode=X
waiter-list
waiter id=process279b098 mode=U requestType=wait
```

Этот результат также позволит получить и другую информацию, которой можно будет воспользоваться в процессе отладки, например, для определения имени пользователя, ставшего жертвой, или блокирующего процесса и другой полезной информации о соединении.

Получение помощи от группы поддержки компании Microsoft

При невозможности решить возникшей в работе SQL Server проблемы, как правило, следует обратиться на следующий уровень поддержки. Если таковым уровнем является поставщик или собственно компания Microsoft, можно попробовать воспользоваться несколькими новыми средствами, которые предназначены для общения со следующим уровнем поддержки. Для лучшей связи с поддержкой можно воспользоваться программами `SQLDumper.exe` и `SQLDiag.exe`, которые позволяют получить общую картину рабочей среды SQL Server и о проблеме, приведшей к возникновению ошибки.

Программа `SQLDumper.exe`

В SQL Server 2000 SP3, для получения дампа окружения SQL Server в случае сбоя была добавлена программа `SQLDumper.exe`. В случае жалоб на зависание сервера организации, осуществляющие поддержку, наподобие службы поддержки Microsoft (Microsoft Product Support Services, PSS), могут также попросить сделать его.

При необходимости создания файла дампа по запросу потребуются идентификатор процесса Windows экземпляра SQL Server. Существует несколько способов получить этот идентификатор. Можно воспользоваться Диспетчером задач и просмотреть регистрационный журнал SQL Server или обратиться к диспетчеру настройки SQL Server (SQL Server Configuration Manager), о котором уже шла речь в этой главе. На странице SQL Server 2005 Services этого диспетчера можно увидеть все службы SQL Server и идентификаторы процессов.

По умолчанию программа `SQLDumper.exe` размещается в папке `C:\Program Files\Microsoft SQL Server\90\Shared`, так как она является общим для всех экземпляров SQL Server, установленных на сервере. Однако эта папка может меняться в зависимости от того, в какую папку устанавливались средства SQL Server. Для того чтобы создать файл дампа для анализа в службе поддержки, необходимо перейти к режиму ввода командной строки, перейдя в папку `C:\Program Files\Microsoft SQL Server\90\Shared`. Находясь в этой папке, можно создать полный дамп или минидамп. Полный дамп занимает значительно больше места, чем минидамп. Так, например, если минидамп займет меньше одного мегабайта, то для размещения полного дампа может потребоваться более 110 Мбайт. Для того чтобы создать полный дамп воспользуйтесь следующей командой.

```
Sqldumper.exe <ProcessID> 0 0x01100
```

Полный дамп будет создан в текущей папке с именем `SQLDmp0001.mdmp`, если программа `SQLDumper.exe` запускалась впервые. После каждого запуска этой программы индекс, используемый в имени файла дампа, будет возрастать. Такой

файл невозможно открыть с помощью обычных текстовых редакторов наподобие Блокнота. Для этого можно будет воспользоваться такими дополнительными средствами, как Visual Studio или одним из средств, отдельно поставляемых службой поддержки. Удобнее работать с минидампом, который содержит самую существенную информацию. Этой информации может оказаться вполне достаточно для службы поддержки. Для создания минидампа следует ввести следующую команду.

```
SqlDumper.exe <ProcessID> 0 0x0120
```

Файл `SQLDUMPER_ERRORLOG.log` можно даже просто просмотреть, чтобы удостовериться в том, что никаких ошибок при создании файла дампа не произошло. Для запуска программы `SQLDumper.exe` необходимо обладать правами локального администратора Windows `SQLDumper.exe` или зарегистрироваться с той же учетной записью, под которой запускалась служба SQL Server.

Программа `SQLDiag.exe`

Средство, которое можно назвать маленьким черным ящиком по сравнению с `SQLDumper.exe` — это программа `SQLDiag.exe`. Те, кто знаком с SQL Server 2000, должны быть знакомы и с программой `PSSDiag.exe`. С ее помощью можно получить файлы с трассировкой SQL и Windows, а также регистрационные журналы настройки. Программа `SQLDiag.exe` унаследовала ее возможности, плюс к старому средству поддержки были добавлены еще и новые возможности. Это средство собирает информацию о вашей системе из:

- системного монитора Windows (`sysmon`);
- журналов регистрации событий Windows;
- трассировок профиля SQL Server;
- журналов регистрации ошибок SQL Server;
- информации о блокировке SQL Server;
- информации о настройке SQL Server.

Так как программа `SQLDiag.exe` собирает слишком много диагностической информации, вам следует запускать ее только в том случае, когда вас об этом попросили или тогда, когда вы готовитесь обратиться в поддержку. Размеры файла трассировки профайлера SQL Server могут расти сами по себе очень быстро, поэтому будьте готовы выводить этот файл на устройство, имеющее много свободного пространства. Этот процесс также использует в своей работе большое количество ресурсов процессора. Это инструментальное средство можно запустить из командной строки или как службу. Для просмотра всех имеющихся в наличии ключей можно воспользоваться ключом `/?`.

Программа `SQLDiag.exe` может принимать конфигурационные файлы как входные. По умолчанию, если имя не задано, этот файл имеет имя `SQLDiag.Xml`. Если конфигурационного XML-файла не существует, он создается с именем `##SQLDiag.XML`. Этот файл можно изменить по своему усмотрению, а затем передать службе поддержки как файл `SQLDiag.XML`.

А теперь, так как вы уже знаете, что может делать программа `SQLDiag.exe`, посмотрим это на примере работы средства с локальным сервером. Если вы не можете работать с сервером непосредственно, вы сможете использовать такое средство, как Terminal Services для удаленного доступа к серверу, так как возможности работы с удаленными серверами у программы `SQLDiag.exe` нет. Для запуска средства перейдите к режиму ввода командной строки. Так как средство работает с пере-

менными среды, вам не обязательно переходить в какую-либо конкретную папку, где размещается сам запускаемый программный файл. Вместо этого в соответствии со сценарием этого примера перейдем в папку C:\Temp или ему подобный, на устройство, которое имеет более 100 Мбайт свободного места.

Стандартным размещением для файлов является папка C:\Program Files\microsoft sql server\90\tools\Binn\SQLDIAG, но его можно изменить с помощью ключа /O. В этом примере введите следующую команду (обратите при этом внимание на отсутствие пробелов после символа "+").

```
sqldiag /B +00:03:00 /E +00:02:00 /OC:\temp /C1
```

Эта команда запустит сбор трассировочной информации программой SQLDiag.exe на протяжении двух-трех минут. Временные установки делаются с помощью ключей /B и /E. Эти два ключа могут использоваться для запуска и остановки диагностики на протяжении 24-х часов. Эта команда также задает программе SQLDiag.exe режим вывода результатов трассировки в папку C:\temp, а ключ /C активизирует сжатие файла средствами Windows. При работе в вашей среде следует подождать, пока не появится подсказка от SQLDiag.exe (она выводится на консоли зелеными буквами) для воспроизводства возникшей проблемы. Полученные результаты будут выглядеть примерно так.

После того как программа SQLDiag.exe отработала, можно переходить в папку C:\temp, заархивировать ее содержимое и отправить полученный архив в Microsoft. В этой папке находится много полезной для группы поддержки информации. Вот некоторые ее составляющие.



```

Microsoft Windows [Version 5.1.2600]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Owner\logfiles
2000/09/18 20:05:50.18 SQLDIAG

It appears that output folder C:\Program Files\Microsoft SQL Server\90\Tools\bin
n\SQLDIAG has been used before. Overwrite the files it contains (Y/N)? y
2000/09/18 20:05:52.76 SQLDIAG Collector version 2000.090.2007.00
2000/09/18 20:05:52.76 SQLDIAG © Microsoft Corp. All rights reserved.

IMPORTANT: Please wait until you see "Collection started" before attempting to
reproduce your issue

2000/09/18 20:05:52.76 SQLDIAG Output path: C:\Program Files\Microsoft SQL Serve
r\90\Tools\bin\SQLDIAG
2000/09/18 20:05:53.01 SQLDIAG Collecting from 1 logical machine(s)
2000/09/18 20:05:53.23 BONIGHT> SQL Server version: 9
2000/09/18 20:05:53.23 BONIGHT> Machine name: BONIGHT (this machine)
2000/09/18 20:05:53.25 BONIGHT> Target machine is not a cluster
2000/09/18 20:05:53.25 BONIGHT> Instance: SQLRSTEST
2000/09/18 20:05:53.25 BONIGHT> Instance: (Default)

2000/09/18 20:06:12.00 SQLDIAG Initialization starting...
2000/09/18 20:06:12.68 BONIGHT> Meta: Get MEMINFO2
2000/09/18 20:06:12.74 BONIGHT> Meta: Get default traces
2000/09/18 20:06:12.82 BONIGHT> Meta: Get SQLDiagver log
2000/09/18 20:06:12.83 BONIGHT> Collecting diagnostic data
2000/09/18 20:06:12.83 SQLDIAG Initialization complete

2000/09/18 20:06:12.84 SQLDIAG Collection started. Press Ctrl+C to stop.
  
```

Рис. 4.13. Результаты работы программы SQLDiag.exe

- ❑ `##files.txt` — перечень файлов из папки `C:\Program Files\Microsoft SQL Server\90\Tools\bin` с датой их создания. Он может использоваться для подтверждения того, что не установлено необходимое обновление;
- ❑ `##envvars.txt` — перечень всех переменных среды данного сервера;
- ❑ `SERVERNAME__sp_sqldiag_Shutdown.OUT` — объединение всех SQL-журналов экземпляра с результатами различных запросов;
- ❑ `log_XX.trc` — группа файлов трассировки профайлера с учетом очень подробной статистики SQL Server;
- ❑ `SERVERNAME_MSINFO32.TXT` — тысячи подробностей о системе и аппаратуре сервера.

Эти файлы могут не только помочь группе поддержки в определении причины сбоя. Эту информацию можно использовать и при проведении регулярных системных работ для определения основных характеристик сервера через определенные моменты времени (перед установкой обновлений, ежемесячно или через произвольные промежутки времени). В таком случае профайлеру программы `SQLDiag.exe` достаточно работать не более нескольких секунд. Такую информацию можно собрать, если это средство работает периодически в режиме моментального снимка. В этом режиме выполняются те же самые функции, но программа завершает работу сразу же после сбора всей необходимой информации. Следующая команда с помощью ключа `/X` переключает программы в режим моментального снимка `SQLDiag.exe`, ключ `/N` (с параметром 2) для создания папки, в которую будут записываться результаты каждого запуска `SQLDiag.exe`.

```
sqldiag /OC:\temp\baseline /X /N 2
```

Папкам присваивается имя `baseline_0000` с последующим инкрементом числовой части. Многие корпорации используют этот режим первого числа каждого месяца, для чего используют SQL Agent или Диспетчер задач, чтобы иметь представление о рабочих характеристиках своего сервера и экземпляра.

Резюме

В этой главе читатели смогли познакомиться с основными концепциями управления SQL Server и устранения неисправностей. Они узнали, как осуществляются настройки SQL Server с помощью средств и сред SQL Server Configuration Manager, SQL Server Surface Area Configuration и, наконец, Management Studio. Мы также показали, как можно управлять подключениями в вашем сервере с использованием инструментального средства Activity Monitor, некоторых основных хранимых процедур и курсоров динамического управления. Кроме того, читатели познакомились с такими средствами управления, как программы `SQLDumper.exe` и `SQLDiag.exe`, которые позволяют отправить основную диагностическую информацию на следующий уровень поддержки или создать архив, хранящий данные с характеристиками эксплуатируемой системы. Теперь во всеоружии знания основных концепций можно перейти к более специальным вопросам эксплуатации SQL Server, таким, например, каким является безопасность. В главе 5 “Автоматизация работы SQL Server” читатель сможет узнать подробнее об автоматизации процессов SQL Server.