

.NET-Betriebshandbuch

(Engl. Originaltitel: [.NET Operations Guide](#))

Whitepaper
von Fernando G. Guerrero

Zusammenfassung

Das Ausführen und Verwalten von Anwendungen und Softwarelösungen auf der Grundlage des Microsoft .NET Framework stellt IT-Profis vor neue Herausforderungen. Eines der grundlegenden Ziele des .NET Framework ist es, einen reibungslosen Betrieb und eine nahtlose Integration vorhandener Windows-Tools zu ermöglichen. Durch die unterschiedlichen Techniken, die beim Ausführen von .NET-Anwendungen eingesetzt werden, können Sie die Leistungsfähigkeit und Zuverlässigkeit des Systems erheblich verbessern. In diesem .NET-Betriebshandbuch werden Informationen und Richtlinien zum Verwalten von Anwendungen und Komponenten basierend auf dem Microsoft .NET Framework sowie Hinweise zu Dokumentationen bereitgestellt, die den Lesern zusätzliche Informationen bieten.

Mit diesem Whitepaper werden die Informationen ergänzt, die in der vollständigen Sammlung der Whitepaper zu .NET-Webdiensten verfügbar sind, einschließlich: *.NET Deployment Guide*, "Securing .NET Web Services", "Interoperation using .NET" und "Performance Tuning a .NET Deployment" (alle englischsprachig).

Dieses Dokument kann als Vorlage zum Verfassen eines Betriebshandbuches für jedes einzelne IT-System verwendet werden, das auf .NET Framework-Lösungen basiert. Das .NET-Betriebshandbuch wurde in erster Linie für IT-Manager und Mitarbeiter des IT-Supports entworfen, um diese sowohl beim Verwalten und Ausführen von Anwendungen und Komponenten, die für das .NET Framework entwickelt wurden, als auch bei der Problembehandlung zu unterstützen. In diesem Handbuch werden jedoch keine umfassenden Informationen zum Verwalten der Windows-Infrastruktur oder bestimmter Produkte des .NET Enterprise Servers bereitgestellt, da dies den Rahmen des Dokuments sprengen würde. Sie können auf der TechNet-Website unter <http://www.microsoft.com/germany/ms/technetdatenbank/overview.asp?siteid=495300> bzw. <http://www.microsoft.com/technet/prodtechnol/windows2000serv/maintain/opsguide/opsguide.asp> (englischsprachig) auf die Reihe der Windows 2000-Betriebshandbücher zugreifen.

MSDN und TechNet stellen Betriebshandbücher für die .NET Enterprise Server-Familie bereit. Auf diese Handbücher können Sie über die entsprechenden Websites unter <http://www.microsoft.com/germany/msdn/> bzw. <http://msdn.microsoft.com> (englischsprachig) und <http://www.microsoft.com/germany/technet/> bzw. <http://www.microsoft.com/technet/default.asp> (englischsprachig) zugreifen. Außerdem können Sie unter <http://www.microsoft.com/technet/prodtechnol/sql/maintain/operate/opsguide/default.asp> (englischsprachig) auf das SQL Server-Betriebshandbuch zugreifen.

Einführung in .NET-Prozesse

Das Verwalten eines auf .NET-Anwendungen und -Komponenten basierenden Computersystems unterscheidet sich nicht wesentlich von der Verwaltung eines anderen IT-Systems. Microsoft Enterprise Services (ES) stellt aufgrund von vorhandenen Kenntnissen im Systementwurf, in der Entwicklung, Bereitstellung und Wartung von Informationssystemen eine Reihe von Umgebungen bereit, die den Richtlinien und optimalen Vorgehensweisen für alle Phasen im Lebenszyklus eines IT-Systems entsprechen. Eine dieser Umgebungen ist das Microsoft Operations Framework (MOF), auf dessen Richtlinien dieses .NET-Betriebshandbuch basiert.

MOF und Enterprise Services

MOF ist eine Zusammenstellung optimaler Vorgehensweisen, Prinzipien und Modelle. Es bietet umfassende technische Unterstützung, um Zuverlässigkeit, Verfügbarkeit, Supportfähigkeit und Verwaltbarkeit für

unternehmenswichtige Produktionssysteme im Zusammenhang mit Lösungen und Diensten zu erzielen, die für Microsoft-Produkte und -Technologien entwickelt wurden.

Jede Umgebung innerhalb der Microsoft ES ist für eine bestimmte Phase der Gültigkeitsdauer von IT-Lösungen bestimmt. Darüber hinaus werden detaillierte Richtlinien für die Beurteilung von Prozessen, Teams und Risiken bereitgestellt, die mit einer speziellen Lebenszyklusphase einer IT-Lösung verbundenen sind.

In Abbildung 1 werden die einzelnen Umgebungen sowie deren Positionierung in die Gesamtstrategie der Enterprise Services dargestellt.

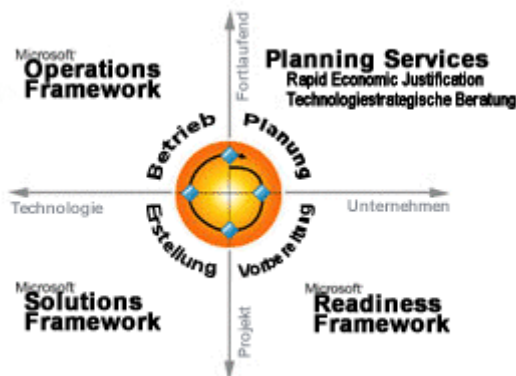


Abbildung 1 Microsoft Enterprise Services Frameworks

MOF konzentriert sich auf die letzte Phase des Lebenszyklus: die erfolgreiche Umsetzung der IT-Lösung in der Produktionsumgebung.

Enterprise Services Frameworks und der IT-Lebenszyklus

Planung

Microsoft-Partner und Microsoft Business Value Services gehören zwar zu keiner bestimmten ES-Umgebung, sie verwenden jedoch bestimmte Tools zum Planen der IT-Infrastruktur, zum Einteilen der Projekte nach Priorität sowie zum Vorbereiten einer allgemeinen Umgebung, um die zahlreichen Aktivitäten eines Projekts erfolgreich umsetzen zu können. Diese Tools liegen in Form von Schulungsunterlagen (Courseware), Whitepapers, Vorlagen und Handbüchern vor. Weitere Informationen zu Microsoft ES finden Sie unter <http://www.microsoft.com/business/services/enterprise.asp> (englischsprachig).

Microsoft stellt offizielle Schulungsunterlagen bereit, die beim Erwerb der erforderlichen Kenntnisse für die Planungsphase einer IT-Lösung hilfreich sind. Diese Kurse beziehen sich zwar nicht speziell auf .NET Framework, die darin vermittelten Techniken sind jedoch auf jede .NET-Lösung anwendbar. Es sind neue Kurse in Vorbereitung, die sich auf die .NET-Lösungen konzentrieren. Die folgenden zwei Kurse beinhalten jedoch Schulungen, die sich direkt auf die Planungsphase von IT-Lösungen beziehen:

- Kurs 1585: "Gathering and Analyzing Business Requirements" (englischsprachig, siehe <http://www.microsoft.com/traincert/syllabi/1585Afinal.asp>)
- Kurs 1515: "Principles of Enterprise Architecture" (englischsprachig, siehe <http://www.microsoft.com/traincert/syllabi/1515Afinal.asp>)

Vorbereitung

Das Microsoft Readiness Framework unterstützt IT-Systemadministratoren bei der Entwicklung individueller und organisationsspezifischer Strategien, um die Verwendung von Microsoft-Produkten und Technologien vorzubereiten. Auch hier werden Microsoft Certified Partner gezielt darauf geschult, Organisationen dabei zu unterstützen, diese Anforderungen durch eine Zusammenarbeit in den einzelnen Prozessen zu erfüllen. Dazu gehören Tools zur Bewertung und Planung der vorbereitenden Maßnahmen, Schulungswegweiser, bereichsbezogene Whitepaper, Unterlagen zum Selbststudium, Kurse, Zertifizierungsprüfungen und Veranstaltungen zum Thema "Vorbereitung des Einsatzes". Zusätzliche Informationen zu Microsoft Readiness Framework finden Sie unter <http://www.microsoft.com/technet/ittasks/plan/teamops/319mrfpm.asp> (englischsprachig).

Microsoft bietet darüber hinaus auch Schulungen für die .NET-Entwicklung in vielen unterschiedlichen Kursen an. Informationen zu diesen Kursen finden Sie unter <http://www.microsoft.com/germany/training/default.htm> bzw. <http://www.microsoft.com/traincert/training/developer/default.asp> (englischsprachig).

Außerdem stellt Microsoft zwei neue Zertifizierungen für .NET-Entwickler bereit:

- Microsoft Certified Application Developer (MCAD) – konzentriert sich auf die Windows- und Webanwendungsentwicklung. Weitere Informationen zu diesem neuen Zertifikat finden Sie unter <http://www.microsoft.com/germany/ms/zertifizierung/program/mcad.htm> bzw. <http://www.microsoft.com/traincert/mcp/mcad/default.asp> (englischsprachig).
- Microsoft Certified Solution Developer (MCS D) – konzentriert sich auf den Entwurf von Systemen auf der Grundlage von .NET-Technologien (für Windows- oder Webplattformen). Umfassende Informationen zu dieser Zertifizierung finden Sie unter <http://www.microsoft.com/germany/ms/zertifizierung/program/mcsd.htm> bzw. http://www.microsoft.com/traincert/mcp/mcad/mcad_mcsd.asp (englischsprachig).

Entwicklung und Bereitstellung

Das Microsoft Solutions Framework bietet Unterstützung für die Planungs-, Entwicklungs- und Bereitstellungsphasen des Projektzyklus. Diese Orientierungshilfen liegen in Form von Whitepapers, Deployment Guides, Fallstudien, Tools, Vorlagen und Schulungsunterlagen in den Bereichen Unternehmensarchitektur, Anwendungsentwicklung, Komponentenentwurf und Infrastrukturbereitstellung vor. Weitere Informationen zum Microsoft Solutions Framework finden Sie unter <http://www.microsoft.com/germany/ms/servicesstrategien/frameworks/msf/> bzw. <http://www.microsoft.com/business/services/mcsmsf.asp> (englischsprachig).

Microsoft bietet die folgenden Kurse im Rahmen des Microsoft Solutions Framework an:

- Kurs 1516: "Principles of Application Development" (englischsprachig, siehe <http://www.microsoft.com/traincert/syllabi/1516Afinal.asp>)
- Kurs 1517: "Principles of Infrastructure Deployment" (englischsprachig, siehe <http://www.microsoft.com/traincert/syllabi/1517Afinal.asp>)
- Kurs 1518: "Principles of Component Design" (englischsprachig, siehe <http://www.microsoft.com/traincert/syllabi/1518Afinal.asp>)

Betrieb

MOF umfasst eine umfassende Sammlung von Betriebsanleitungen von IT-Systemen, die in Form von Whitepapers, Betriebshandbüchern, Bewertungstools, Kits, optimalen Vorgehensweisen, Fallstudien und Supporttools für Benutzer, Prozesse und Technologien bereitgestellt werden, um Produktionssysteme in einer komplexen, verteilten IT-Umgebung effizient verwalten zu können.

Microsoft bietet den folgenden Kurs im Rahmen des MOF an:

- Kurs 1737: "Microsoft Operations Framework Essentials" (englischsprachig, siehe <http://www.microsoft.com/traincert/syllabi/1737Afinal.asp>)

.NET-Prozesse im .NET-Lebenszyklus

Das MOF-Prozessmodell besteht aus vier eng miteinander verknüpften Bereichen systembetriebsorientierter Aktivitäten. Diese MOF-Quadranten umfassen Folgendes:

- Änderung
- Betrieb
- Support
- Optimierung

Jeder einzelne Quadrant erfüllt einen speziellen Dienstzweck, der durch Ausführen bestimmter Dienstverwaltungsfunktionen erreicht wird. Dies geht aus dem nächsten Abschnitt dieses Dokuments hervor. Jeder Quadrant beinhaltet eine Reihe von Dienstverwaltungsfunktionen (Service Management Functions oder SMFs). Die Definitionen dieser Dienste und Funktionen werden im MOF-Prozessmodell beschrieben. Es ist wichtig, die Rollen und Verantwortungsbereiche zum Durchführen der im Prozessmodell beschriebenen Aktivitäten zu definieren. Das MOF-Teammodell unterstützt Sie beim Definieren dieser Rollen und Verantwortungsbereiche.

Der .NET-Lösungsprozess für den Betrieb beginnt mit der Konfiguration der an der jeweiligen .NET-Lösung beteiligten Systeme und Anwendungen. Der Quadrant "Betrieb" deckt die fortlaufenden Aktivitäten bei der Ausführung und Verwaltung einer .NET-Lösung ab. Supportvorgänge beinhalten mehrere Prozesse zum Identifizieren von Problemen und potenziellen Verbesserungen im System.

Die beim Betrieb (Verwalten) und Support einer .NET-Anwendung gesammelten Erfahrungen stellen ein äußerst hilfreiches Feedback bereit, das zum Verbessern und Optimieren des Anwendungsentwurfs genutzt werden kann. Dieser neue Entwurf führt zu einer optimierten Anwendung, die noch effizienter verwaltet und unterstützt werden kann. Aus diesem Grund sollten diese Quadranten als dynamischer Kreis betrachtet werden, bei dem das System bei jeder neuen Iteration weiter verbessert wird.

Zum Vervollständigen des MOF-Prozessmodells müssen Sie bestimmte Verwaltungsüberprüfungen zum Beurteilen der Effizienz jedes Quadranten bereitstellen. Die Verwaltungsüberprüfungen umfassen Folgendes:

- Funktionsbereitschaft der Version
- Betrieb
- Vereinbarung auf Dienstebene (Service Level Agreement, SLA)
- Genehmigte Version

Sie können die im MOF-Prozessmodell beschriebenen Richtlinien für die Organisation der Vorgänge bei der Bereitstellung und Verwaltung von .NET-Lösungen nutzen. Dies umfasst beispielsweise folgende Punkte:

- Allgemeiner Betrieb einer .NET-Lösung
- Zertifizierungsprozess einer .NET-Lösung
- Beurteilung der Benutzerakzeptanz
- Stagingprozess (Vorbereiten) einer .NET-Lösung, von der Testumgebung bis zur endgültigen Produktionsumgebung
- Integration in andere Systeme und Dienste

.NET-Dienstverwaltungsfunktionen

Jeder Quadrant beinhaltet unterschiedliche Aufgaben. In den allgemeinen MOF-Richtlinien werden diese Aufgaben als Auflistung von Dienstverwaltungsfunktionen (Service Management Functions oder SMFs) für jeden Quadranten definiert. Die meisten dieser SMFs beziehen sich direkt auf die Verwaltung einer .NET-Anwendung. Es folgen ein paar Beispiele dafür:

- Configuration Management
- Security Administration
- Job Scheduling
- Service Desk
- Incident Management
- Capacity Management

Weitere Informationen zu den Dienstverwaltungsfunktionen finden Sie unter <http://www.microsoft.com/germany/ms/technetdatenbank/overview.asp?siteid=495299> bzw. <http://www.microsoft.com/business/services/MOFpmdetail.asp> (englischsprachig).

Rollen und Verantwortungsbereiche für .NET-Vorgänge

Im MOF-Teammodell werden Rollen sowie die jeder Rolle zugeordneten Verantwortungsbereiche definiert, um die Ziele der einzelnen SMF zu erreichen. Jede Rolle kann an mehreren SMFs beteiligt sein, und jeder Quadrant erfordert die Mitwirkung mehrerer Rollen.

Beim Verwalten eines auf .NET-Anwendungen, -Komponenten und -Diensten basierenden Systems können folgende Rollen identifiziert werden:

- Release-Rolle: Release Manager, Kommunikationskoordinator, Dokumentationskoordinator, Rolloutkoordinator, Testkoordinator, Konfigurationsmanager
- Betriebsrolle: Netzwerkmanager
- Sicherheitsrolle: Sicherheitsmanager
- Supportrolle: Service Desk-Manager
- Infrastrukturrolle: Schulungsleiter, Availability Manager, Capacity Manager

Eine klare Definition von Rollen und Verantwortungsbereichen ist für den effizienten Betrieb eines IT-Systems von großer Bedeutung. Informationen zu den Verantwortungsbereichen dieser Rollen finden Sie in der offiziellen MOF-Dokumentation unter

<http://www.microsoft.com/germany/ms/technetdatenbank/overview.asp?siteid=495299> bzw. <http://www.microsoft.com/business/services/MOFteam.asp> (englischsprachig).

Spezifisches schichtbasiertes Systembetriebsmanagement in einer .NET-Lösung

Das Verwalten eines auf .NET-Anwendungen und Komponenten basierenden Systems umfasst viele verschiedene Aufgaben, die je nach Systemschicht unterschiedlich ausfallen können:

- Netzwerkinfrastrukturverwaltung
- Datenspeicherverwaltung
- ASP.NET-Anwendungsverwaltung
- .NET-Webdienste

In den nachfolgenden Abschnitten dieses Dokuments werden einige dieser Schichten im Einzelnen beschrieben. Das .NET Framework kann zum Implementieren von eigenständigen und Netzwerkanwendungen unter Windows verwendet werden. Ihre Ausführung unterscheidet sich kaum von der Ausführung von Windows-Standardanwendungen. Das .NET Framework weist jedoch auch ein paar entscheidende Unterschiede auf, die in

den folgenden Abschnitten dieses Dokuments detaillierter erläutert werden. Mithilfe dieser Richtlinien sollten Sie in der Lage sein, ein Betriebshandbuch für Ihre spezielle .NET-Lösung zu verfassen.

Verwenden des Systemmonitors zum Überwachen von .NET-Anwendungen

.NET Framework stellt mehrere Objekte und Leistungsindikatoren bereit, die Sie mithilfe des Systemmonitors überwachen können. Eine vollständige Liste der in .NET-Anwendungen verwendeten Leistungsindikatoren finden Sie unter <http://msdn.microsoft.com/library/en-us/cpgenref/html/gngrfperformancecounters.asp> (englischsprachig).

Mithilfe dieses Tools können Sie grafisch anzeigen, wie die Anwendung die Ressourcen im System verwendet, z. B. die Arbeitsspeichernutzung und die Anzahl der ausgelösten Ausnahmen.

Sie können dieses Tool über das Startmenü starten. Sie sollten jedoch beachten, dass sich die zugehörige Menüstruktur nach dem installierten Betriebssystem richtet. Als sinnvolle Alternative können Sie das Tool direkt über ein Eingabeaufforderungsfenster ausführen, oder Sie gehen folgendermaßen vor:

1. Klicken Sie auf die Schaltfläche **Start**, um das Startmenü anzuzeigen, und wählen Sie **Ausführen (Run)** aus.
2. Geben Sie **Perfmon.exe** ein (siehe Abbildung 2), und klicken Sie auf **OK**.

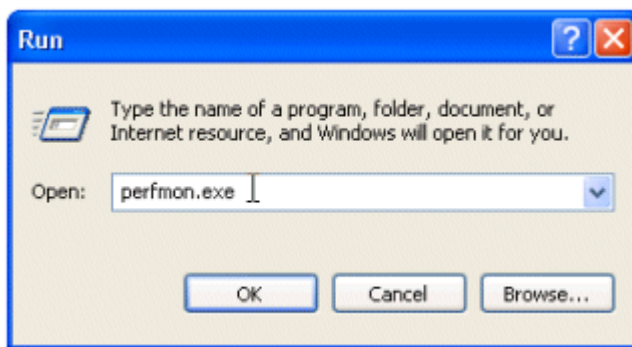


Abbildung 2 Starten des Systemmonitors über das Startmenü

Hier wird der Systemmonitor nur mit den Standardindikatoren angezeigt.

3. Klicken Sie auf der Symbolleiste auf das Symbol **Hinzufügen** (das Pluszeichen), und wählen Sie den anzuzeigenden Leistungsindikator aus (die .NET-Leistungsindikatoren werden am Listenanfang angezeigt). Siehe auch Abbildung 3.

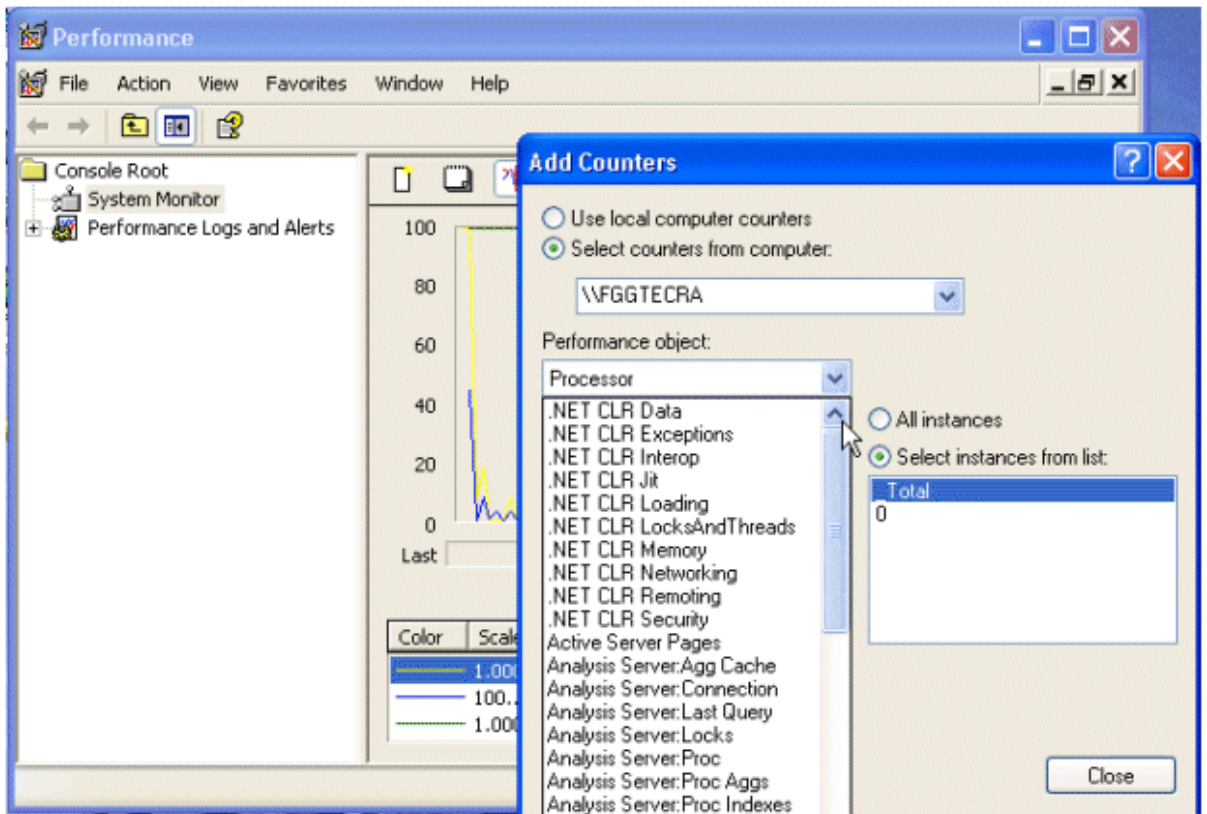


Abbildung 3 Hinzufügen von .NET-Leistungsindikatoren im Systemmonitor

.NET-Assistenten

.NET Framework stellt drei Assistenten bereit, die Sie beim Durchführen allgemeiner Aufgaben unterstützen. Diese Assistenten sind Bestandteil des Tools **ConfigWizards.exe**. Dieses Tool befindet sich im Verzeichnis `%Windir%\Microsoft.NET\Framework\%.NETversion%` (z. B. `C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\ConfigWizards.exe`).

Auf dieses Tool können Sie über das Startmenü als Teil der Gruppe **Verwaltung** zugreifen. Nach dem Starten des Programms wird das folgende Dialogfeld angezeigt (siehe Abbildung 4).

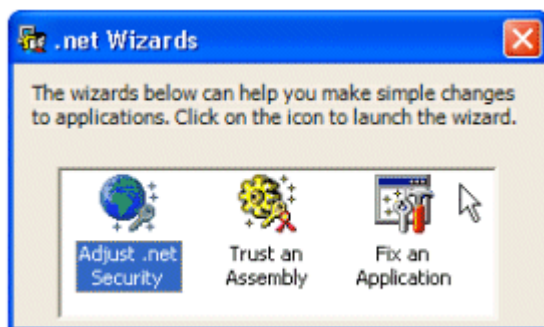


Abbildung 4: Die .NET-Assistenten (.net Wizards)

.NET-Sicherheit anpassen

Mithilfe des ersten Assistenten (.NET-Sicherheit anpassen) können Sie die für jede Zone geltenden Sicherheitsstufen entweder für den aktuellen Benutzer oder für den gesamten Computer anpassen. Diese Einstellungen haben Einfluss darauf, wie .NET-Anwendungen auf Ressourcen aus dem Internet, dem lokalen Intranet oder auf andere Ressourcen zugreifen können, die auf demselben Computer oder im lokalen Netzwerk verfügbar sind. Beim Verwenden dieses Tools müssen Sie zuerst die anzupassenden Einstellungen auswählen (lokaler Computer oder aktueller Benutzer). Anschließend wird die bekannte Benutzeroberfläche des Internet Explorer mit den Internetereinstellungen angezeigt, so dass Sie die Sicherheitsstufe für jede Zone anpassen können (siehe Abbildung 5).



Abbildung 5 Anpassen der Sicherheitsstufen für jede Zone (Adjust the Security Level for Each Zone)

Einer Assembly vertrauen

Mithilfe dieses Tools können Sie die Sicherheitsstufe für eine bestimmte Assembly anpassen. Führen Sie zu diesem Zweck die folgenden Schritte durch:

1. Klicken Sie auf das Symbol **Einer Assembly vertrauen (Trust an Assembly)**.
2. Wählen Sie entweder Änderungen am lokalen Computer oder am lokalen Benutzer aus, und klicken Sie dann auf **Weiter (Next)**.
3. Geben Sie den Pfad oder URL der zu vertrauenden Assembly ein, und klicken Sie dann auf **Weiter**.
4. Wählen Sie die Vertrauensstufe für diese Assembly aus (siehe Abbildung 6). Beachten Sie, dass durch die Auswahl von **Voll vertrauenswürdig (Full Trust)** für eine bestimmte Assembly diese dann über Vollzugriff auf alle Ressourcen verfügt. Lesen Sie beim Verschieben des Schiebereglers die Beschreibung für jede Stufe durch.

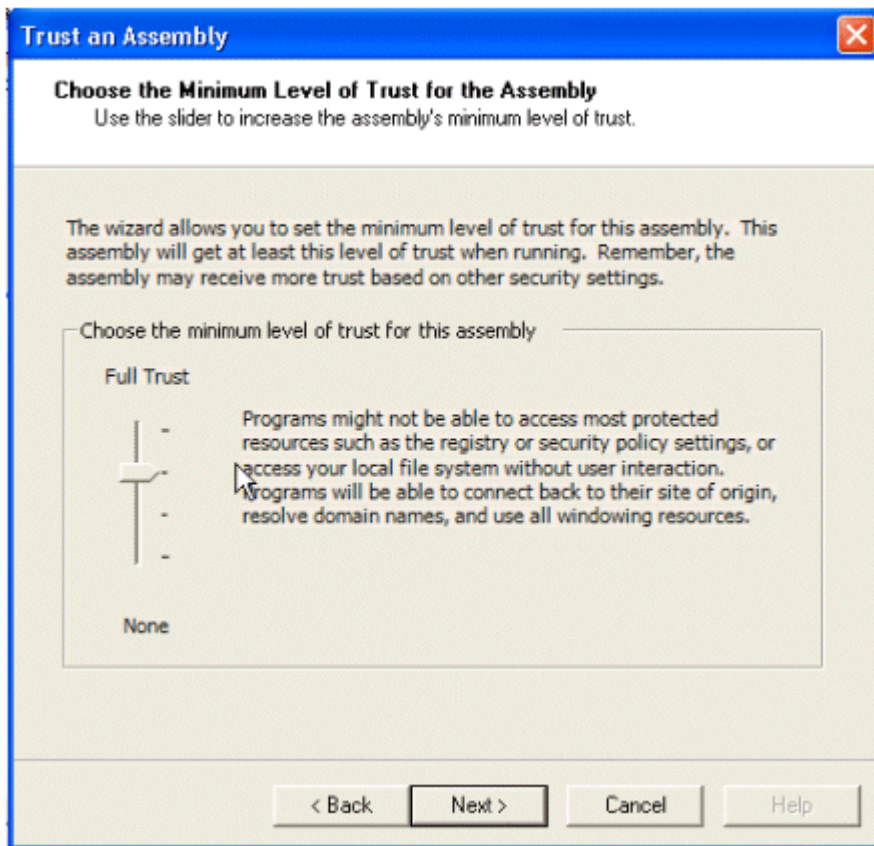


Abbildung 6 Einer Assembly vertrauen (Trust an Assembly)

5. Klicken Sie auf der Bestätigungsseite auf **Fertig stellen**, oder gehen Sie zur vorherigen Seite zurück, um die Vertrauensstufe anzupassen.

Bei dieser Methode handelt es sich um einen eher groben Ansatz, den Sie daher nur als Notlösung in Erwägung ziehen sollten. Das Verwenden der richtigen, auf Sicherheitsaspekten beruhenden Codierungstechniken ist als bevorzugte Methode anzusehen. Auf diese Weise können Sie die verfügbaren Funktionen für bestimmte Benutzer und Benutzergruppen umfassend überwachen. Wenn Sie jedoch eine bestimmte Anwendung ausführen müssen und nicht über die entsprechenden Zugriffsrechte zum Ändern der codebasierten Sicherheitssteuerung verfügen, kann das Problem mithilfe dieses Tools sehr schnell behoben werden.

Eine Anwendung reparieren

Mit dem in Abbildung 7 dargestellten .NET-Anwendungswiederherstellungstool können Sie die zu einem bestimmten Zeitpunkt gespeicherten Anwendungsrichtlinieneinstellungen wiederherstellen oder der Anwendung zu weiteren Testzwecken den abgesicherten Modus (**Sicherer Anwendungsmodus (Application Safemode)**) zuweisen. Sie sollten jedoch bedenken, dass bei einer Anwendung im abgesicherten Modus Sicherheits- oder Verhaltensprobleme auftreten könnten, da die Anwendung in diesem Modus sämtliche Richtlinien des Verlegers ignoriert.

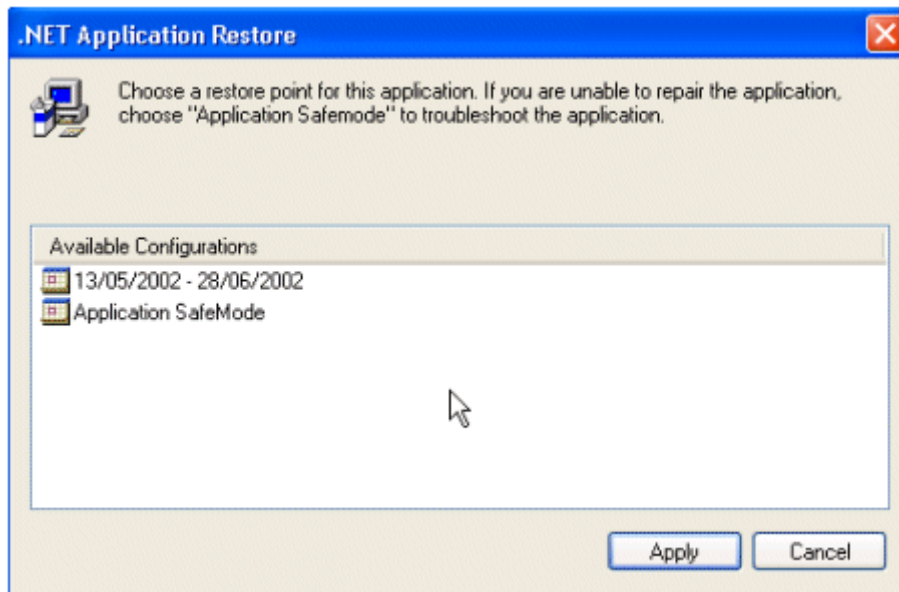


Abbildung 7 Wiederherstellen der Anwendungskonfigurationseinstellungen

.NET-Konfiguration

Bei diesem Tool handelt es sich um ein MMC-Snap-In (Microsoft Management Console) mit der Bezeichnung **mscorcfg.msc**. Es befindet sich im Standardverzeichnis des .NET Framework:

C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\mscorcfg.msc, und Sie können über die Menügruppe **Verwaltung** auf dieses Tool zugreifen.

Mit dem Tool können Sie die folgenden Aufgaben ausführen:

- Assemblys
 - Verwalten von Assemblys im globalen Assemblycache
 - Konfigurieren von Assemblys im globalen Assemblycache
- Remoting
 - Anpassen von Remotediensten
- Sicherheit
 - Anzeigen der aktuellen .NET Framework-Sicherheitskonfiguration
 - Erhöhen der Assemblyvertrauensstellungen
 - Anpassen der Zonensicherheit
 - Auswerten einer Assembly
 - Erstellen eines Bereitstellungspakets
 - Zurücksetzen aller Richtlinien Ebenen
- Einzelne Anwendungen
 - Anzeigen der zuvor mithilfe dieses Tools konfigurierten Anwendungen
 - Hinzufügen einer mit diesem Tool zu konfigurierenden Anwendung
 - Anzeigen der Eigenschaften einer Anwendung
 - Anzeigen der Assemblyabhängigkeiten einer Anwendung
 - Konfigurieren einer Assembly für eine Anwendung
 - Anpassen von Remotediensten für eine Anwendung

Detaillierte Informationen zu diesem Tool und zum schrittweise Ausführen der einzelnen Aufgaben finden Sie unter <http://msdn.microsoft.com/library/en-us/cptools/html/cpconnetframeworkadministrationtoolmscorcfgmsc.asp> (englischsprachig)

In Abbildung 8 wird die gesamte Struktur mit allen verfügbaren Optionen des Tools angezeigt.



Abbildung 8 Grundlegende Aufgaben im Tool .NET Framework-Konfiguration

Verwalten der .NET-Netzwerkinfrastruktur

Das Betriebshandbuch für eine .NET-Lösung sollte mindestens zwei Diagramme umfassen:

- Ein **physisches Netzwerktopologiediagramm**, aus dem die Verteilung der für die Unterstützung der .NET-Lösung genutzten Netzwerkkomponenten hervorgeht.
- Ein **logisches Flussdiagramm**, um alle Softwarekomponenten perspektivisch zu platzieren. In diesem Diagramm wird eine andere Perspektive der Netzwerktopologie bereitgestellt. Beide Ansichten sollten jedoch genügend Verbindungen untereinander aufweisen, um ihre Beziehung zu verdeutlichen. Das Hauptziel dieses Diagramms besteht darin, die unterschiedlichen Komponenten und die vorhandenen Abhängigkeiten aufzuzeigen.

In dem Betriebshandbuch sollte der logische Entwurf des Netzwerkes, einschließlich der internen Infrastruktur und der Verbindungen mit dem Internet, wie folgt beschrieben sein:

- DNS-Infrastruktur
- IP-Konfiguration für alle Server

Die Internetnetzwerkconfiguration sollte so definiert sein, dass nur die Clients und Protokolle unterstützt werden, die tatsächlich zum Ausführen der vorgesehenen Dienste erforderlich sind. Das Aktivieren zusätzlicher unnötiger Protokolle und Dienste führt zu einer erhöhten Belastung des Servers und bringt keine weiteren Vorteile für die Benutzer mit sich.

Von einer detaillierten Ausführung der wichtigsten Netzwerkvorgänge wird in diesem Dokument abgesehen. Zusätzliche Informationen und spezielle Richtlinien für Netzwerkprozesse finden Sie im *Network Administration-Betriebshandbuch* unter

<http://www.microsoft.com/germany/ms/technetdatenbank/overview.asp?siteid=448881> bzw.

<http://www.microsoft.com/technet/prodtechnol/windows2000serv/maintain/opsguide/netadmog.asp>

(englischsprachig).

Verwalten von SQL Server

Microsoft SQL Server ist ein wesentlicher Bestandteil der meisten .NET-Lösungen, und eine nicht ordnungsgemäße Verwaltung von SQL Server kann sich auf die Gesamtleistung der .NET-Lösung auswirken. SQL Server verwendet einen besonderen Dienst, den SQL Server-Agenten, um die meisten sich wiederholenden automatisierten administrativen Vorgänge in SQL Server durch Definieren von Operatoren, Warnungen und Aufträgen auszuführen. Das SQL Server-Auftragsmodul stellt eine umfassende Programmierumgebung bereit, in der Sie komplexe Aufträge mithilfe der Transact-SQL-Sprache sowie mit anderen auf dem Server verfügbaren Skriptsprachen (in der Regel VBScript und JScript) verfassen können.

SQL Server stellt eigene Sicherungsfunktionen bereit, um die Systemsicherheit und -wiederherstellung zu garantieren. Darüber hinaus sollten Sie sicherstellen, dass auch die vorgesehene Sicherungsstrategie eine Systemwiederherstellung garantiert. Eine vernünftige Sicherungsstrategie ist nur so gut wie der Testprozess, der angewendet wird, um kontinuierlich zu überprüfen, ob die Wiederherstellungsverfahren erwartungsgemäß ausgeführt werden.

Von einer detaillierten Beschreibung der SQL Server-Vorgänge wird in diesem Dokument abgesehen.

Zusätzliche Informationen zu allgemeinen SQL Server-Vorgängen finden Sie im *SQL Server 2000-*

Betriebshandbuch unter <http://www.microsoft.com/germany/ms/technetdatenbank/overview.asp?siteid=515505>

bzw. <http://www.microsoft.com/technet/prodtechnol/sql/maintain/operate/opsguide/default.asp>

(englischsprachig).

Verwalten von ASP.NET-Anwendungen

ASP.NET stellt ein umfassendes und flexibles Konfigurationssystem bereit, mit dem das Verwalten eines .NET-basierten Systems vereinfacht wird. Sämtliche Konfigurationsinformationen werden in XML-basierten Textdateien gespeichert, so dass Sie problemlos jeden Texteditor verwenden können, um diese Dateien zu ändern, ohne die Anwendungen neu codieren zu müssen.

Jedes Anwendungsverzeichnis kann über eine eigene Konfigurationsdatei (mit der Bezeichnung **web.config**) verfügen, in der die Konfigurationseinstellungen für die in diesem Verzeichnis und in allen untergeordneten Verzeichnissen gespeicherten ASP.NET-Anwendungen verwaltet werden. Jede Konfigurationsdatei in einem untergeordneten Verzeichnis kann Konfigurationseinstellungen zu den im übergeordneten Verzeichnis verfügbaren hinzufügen. Eine Konfigurationsdatei kann auch Einstellungen außer Kraft setzen, um eine bestimmte Einstellungsvererbung zu blockieren.

Über eine besondere Konfigurationsdatei (mit der Bezeichnung **machine.config**) werden Konfigurationseinstellungen für den gesamten Webserver bereitgestellt. Diese Datei befindet sich im folgenden Verzeichnis: **%runtime install path%\Config\Machine.config**. Im folgenden Beispiel lautet der Speicherort

dieser Datei **C:\WINDOWS\Microsoft.NET\Framework\v1.0.3705\CONFIG\machine.config**, wobei **v1.0.3705** die installierte .NET-Laufzeitversion darstellt.

Anhand des folgenden Beispiels wird veranschaulicht, wie ASP.NET die Vererbung der Konfiguration anwendet.

1. Ein Benutzer startet die folgende ASP.NET-Anwendung:

http://www.sqlclient.net/SalesReports/Year2002/SalesJune.aspx

2. ASP.NET durchsucht das virtuelle Verzeichnis **SalesReports**, um den physischen Speicherort dieser Datei zu ermitteln.
3. ASP.NET sucht nach einer Datei mit dem Namen **web.config**, die sich in diesem physischen Verzeichnis befindet.
4. Anschließend sucht ASP.NET nach einer Datei mit dem Namen **web.config**, die sich im Unterverzeichnis **Year2002** befindet. Wenn diese Datei vorhanden ist, werden die Konfigurationseinstellungen mit den Einstellungen aus dem vorherigen Schritt zusammengeführt.
5. Im Anschluss daran werden diese Konfigurationsinformationen von ASP.NET für nachfolgende Ausführungen derselben Anwendung oder einer anderen Anwendung zwischengespeichert, die sich im logischen Verzeichnis befindet.

Beachten Sie, dass die Konfigurationseinstellungen anders ausfallen können, wenn ein anderer Benutzer den folgenden URL anfordert: **http://www.sqlclient.net/GeneralReports/SalesReports/Year2002/SalesMay.aspx**, und wenn dieses Verzeichnis **Year2002** demselben physischen Verzeichnis aus dem vorherigen Schritt entspricht. Der Grund dafür ist in diesem Fall, dass ASP.NET die Konfigurationseinstellungen in den Dateien mit der Bezeichnung **web.config** auf bestimmten Ebenen zusammenführen muss. Diese umfassen Folgendes:

- Das physische Verzeichnis, auf das das virtuelle Verzeichnis **GeneralReports** verweist.
- Das Unterverzeichnis **SalesReports**.
- Das Unterverzeichnis **Year2000**.

ASP.NET erkennt die an den Konfigurationsdateien vorgenommenen Änderungen und wendet automatisch neue Konfigurationseinstellungen auf die von den Änderungen betroffenen Webressourcen an. Der Server muss nicht neu gestartet werden, damit die Änderungen in Kraft treten. Hierarchische Konfigurationseinstellungen werden automatisch neu berechnet und zwischengespeichert, wenn Änderungen an einer Konfigurationsdatei in der Hierarchie vorgenommen werden. Eine Ausnahme bildet hierbei der Abschnitt `<processModel>`.

ASP.NET konfiguriert den Microsoft Internet Information Server (IIS), um speziell das Durchsuchen dieser Konfigurationsdateien zu verhindern. Darüber hinaus sollten Sie sicherstellen, dass keine anderen Zugriffspfade zu diesen Dateien vorhanden sind, um Sicherheitsrisiken zu vermeiden.

In der Begleitdokumentation für die Bereitstellung dieser .NET-Reihe wird eine vollständige Beschreibung der unterschiedlichen Konfigurationseinstellungen in Dateien mit der Bezeichnung **web.config** bereitgestellt. Zusätzliche Informationen zu den ASP.NET-Konfigurationsdateien finden Sie unter <http://msdn.microsoft.com/library/en-us/cpguide/html/cpconaspnetconfiguration.asp> (englischsprachig).

Überwachen von ASP.NET-Anwendungen

ASP.NET bietet einzelne Leistungsindikatoren pro Anwendung sowie globale Leistungsindikatoren zum Überwachen des gesamten ASP.NET-Systems.

Zum Anzeigen dieser Objekte können Sie die folgenden Schritte durchführen:

1. Führen Sie die Anwendung **perfmon.exe** (wie bereits weiter oben erläutert) aus.
2. Klicken Sie auf der Symbolleiste auf das Symbol **Bericht anzeigen** (ist mit dem Editor-Symbol vergleichbar), oder drücken Sie STRG+R.
3. Klicken Sie auf das Symbol **Hinzufügen** (das Pluszeichen), oder drücken Sie STRG+I.

4. Wählen Sie aus der Dropdownliste **Leistungsobjekt (Performance object)** die Option **ASP.NET Applications** aus.
5. Aktivieren Sie **Alle Leistungsindikatoren (All counters)** oder **Leistungsindikatoren wählen (Select counters from list)**, wenn für Sie nur bestimmte Leistungsindikatoren von Interesse sind.
6. Jetzt können Sie entweder bestimmte ASP.NET-Anwendungen zum Überwachen auswählen (**Instanzen wählen (Select instances from list)**) oder die gesamten Leistungsindikatoren sämtlicher ASP.NET-Anwendungen abrufen (**Alle Instanzen (All instances)**). Klicken Sie auf die Schaltfläche **Hinzufügen (Add)**, um die ausgewählten Indikatoren zum Bericht hinzuzufügen (siehe Abbildung 9).

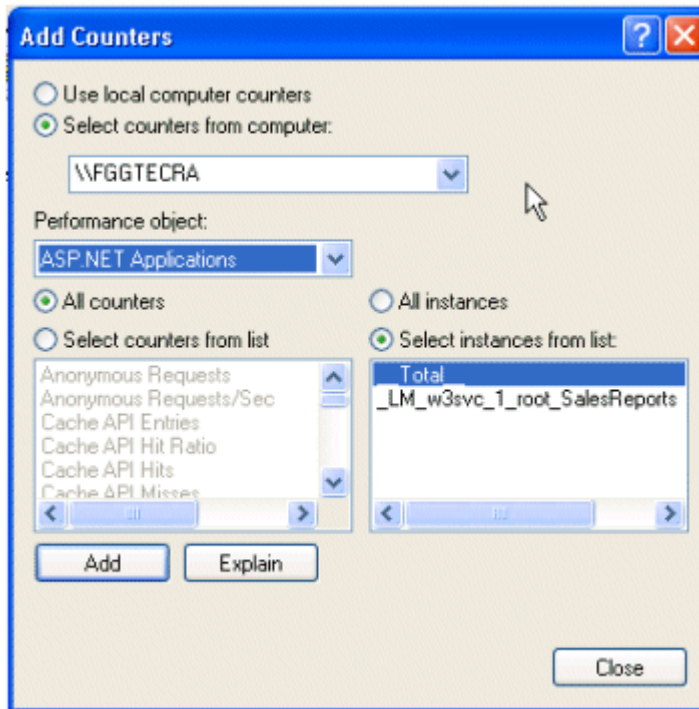


Abbildung 9 Hinzufügen von Leistungsindikatoren für ASP.NET-Anwendungen zum Systemmonitor

7. Sie können Leistungsindikatoren hinzufügen, die sich auf das gesamte ASP.NET-System beziehen (über das **ASP.NET-Datenobjekt**), oder Leistungsindikatoren für bestimmte ASP.NET-Anwendungen auswählen (über das **ASP.NET Applications**-Leistungsobjekt). Wenn Sie das **ASP.NET Applications**-Objekt auswählen, können Sie angeben, welche Anwendung überwacht werden soll. Sie können auch die gesamten Leistungsindikatoren für sämtliche ASP.NET-Anwendungen abrufen (mithilfe der Instanz **_Total_**). Wenn Sie jedoch das **ASP.NET**-Leistungsobjekt auswählen, können Sie keine bestimmte Anwendungsinstanz auswählen, weil diese Indikatoren global über das ASP.NET-System abgerufen werden (siehe Abbildung 10).

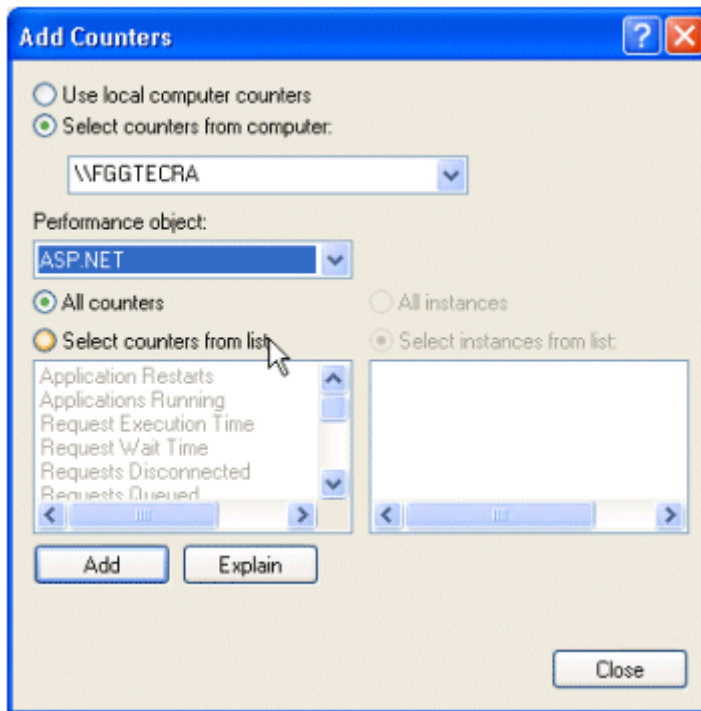


Abbildung 10 Hinzufügen von ASP.NET-Leistungsindikatoren zum Systemmonitor

Wenn mehrere Versionen des .NET Framework auf dem System installiert sind, sollten Sie beachten, dass Sie die Leistungsindikatoren der entsprechenden Versionen auswählen können, wie aus Abbildung 11 hervorgeht. Hier sind zwei unterschiedliche Versionen (1.0.3215.14 & 1.0.3705.0) vorhanden. In diesem Fall ist das **ASP.NET**-Datenobjekt identisch mit **ASP.NET v1.0.3705.0**, und das **ASP.NET Applications**-Datenobjekt stimmt mit **ASP.NET Apps v1.0.3705.0** überein.

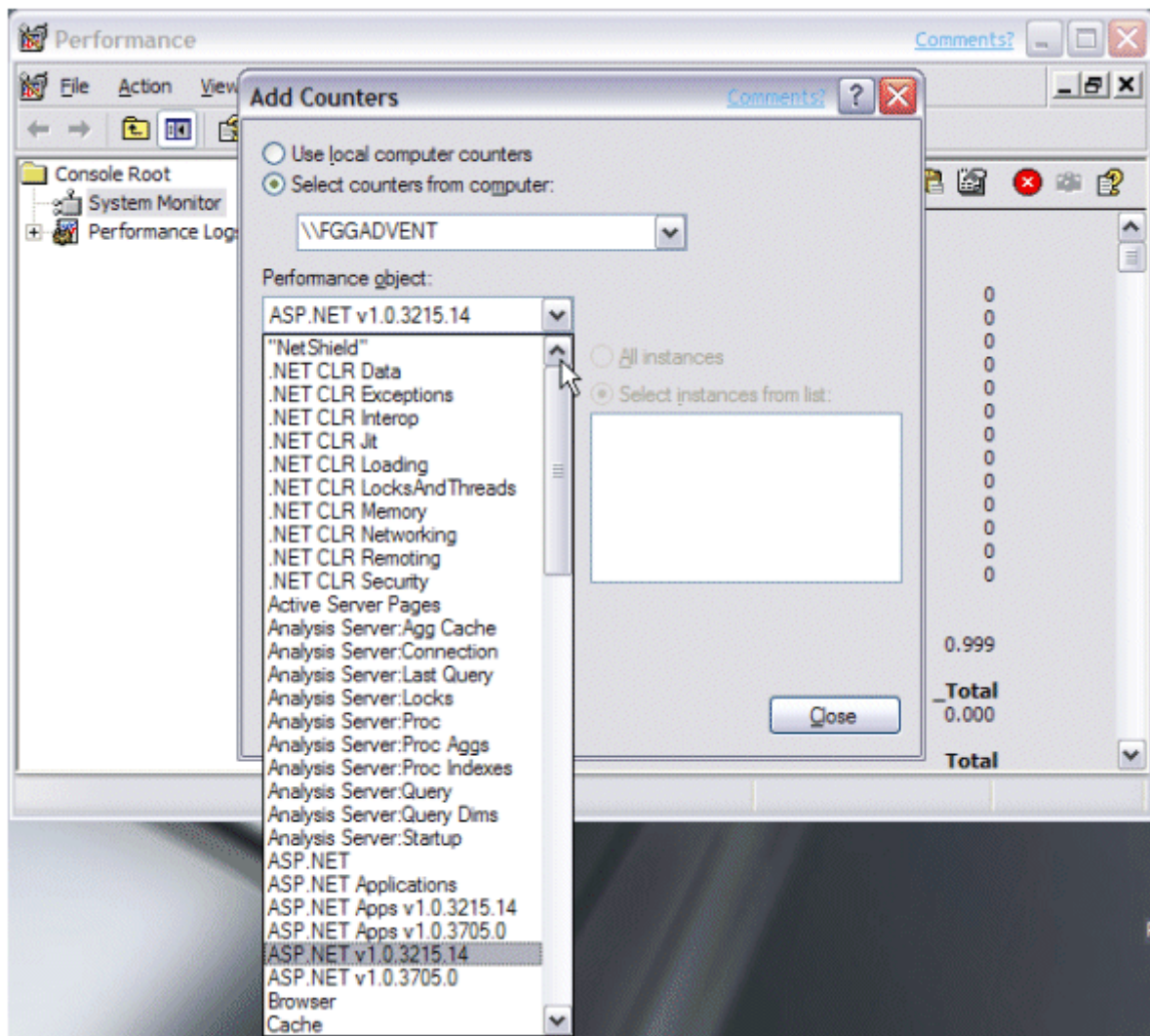


Abbildung 11 Über den Systemmonitor bereitgestellte .NET-Datenobjekte

Klicken Sie nach dem Auswählen der Datenobjekte und Leistungsindikatoren auf **Schließen (Close)**, dann wird der in Abbildung 12 dargestellte Bericht angezeigt.

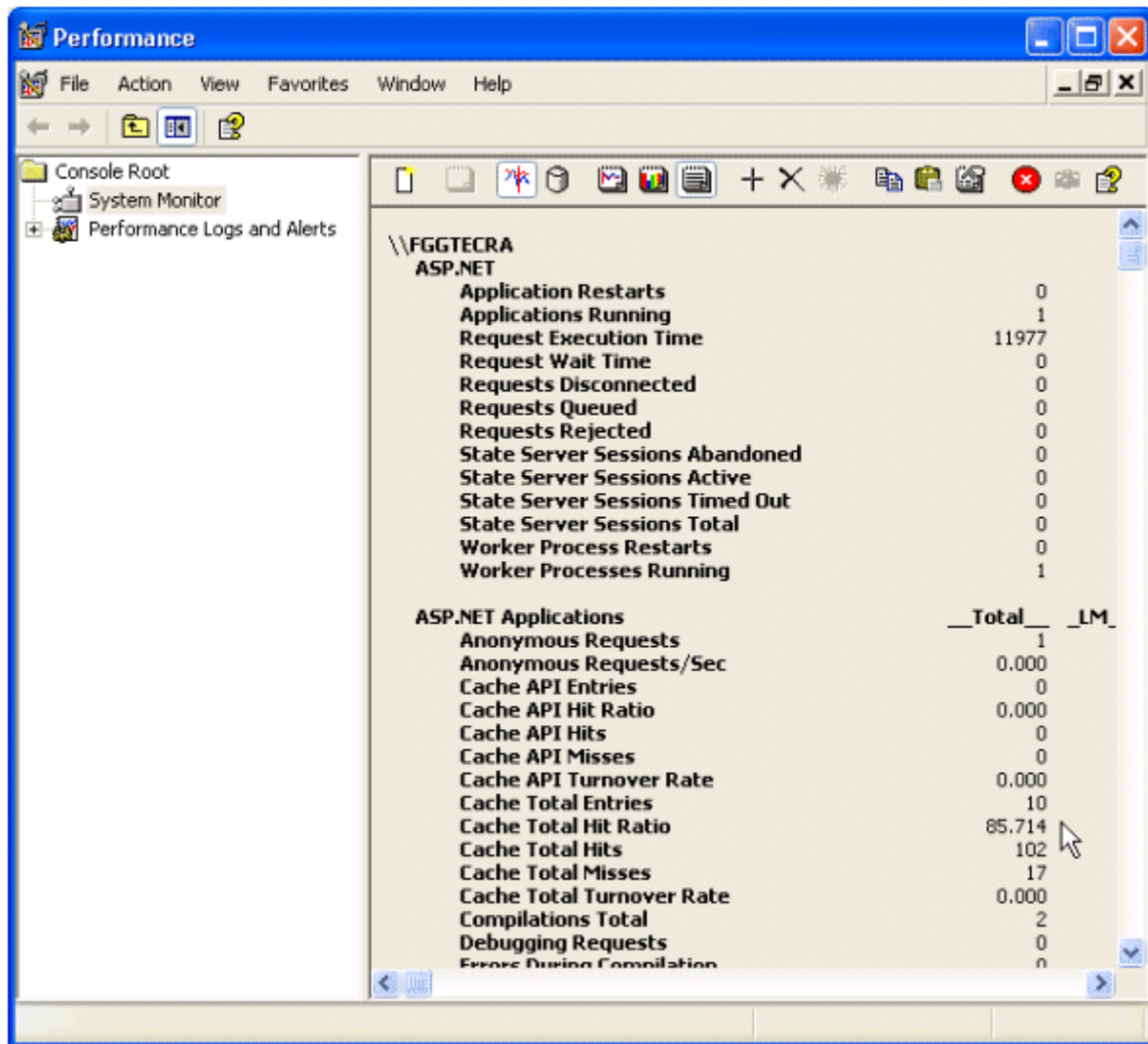


Abbildung 12 Auflisten von .NET-Leistungsindikatoren im Systemmonitor

Durch Auswählen anderer Symbole auf der Symbolleiste können Sie diese Informationen als Diagramm (STRG+G) oder als Histogramm (STRG+B) anzeigen.

Im Folgenden werden einige besonders zu beachtende Leistungsindikatoren aufgeführt:

- **Physikalischer Datenträger\Lesevorgänge/s** und **Physikalischer Datenträger\Schreibvorgänge/s**. Hierbei handelt es sich um allgemeine Indikatoren, die jedoch wichtige Informationen bereitstellen, mit denen physische Engpässe im Speichersystem erkannt werden können.
- **Prozess: ASP.NET\Prozessorzeit (%)**. Hohe Werte dieses Indikators können darauf hinweisen, dass die Arbeitsauslastung von ASP.NET auf mehrere Server (beispielsweise in einer Webfarm) aufgeteilt werden sollte. Beachten Sie, dass dieser Wert durch die Anzahl an Prozessoren eines Servers mit mehreren Prozessoren dividiert werden sollte, um die tatsächliche Arbeitsauslastung pro Prozessor zu ermitteln.
- **Prozess: ASP.NET\Private Bytes**. Dieser Indikator sollte zusammen mit dem Indikator **Ausgeführte Anwendungen** überwacht werden. Bei einigen ASP.NET-Anwendungen könnten Arbeitsspeicherverluste auftreten, wenn sich die Anzahl der Anwendungen nicht verändert, der Wert des Indikators **Private Bytes** jedoch im Laufe der Zeit steigt.
- **ASP.NET Applications\Anforderungen/Sek**. Mit diesem Indikator können Sie die tatsächliche Arbeitsauslastung des ASP.NET-Subsystems genau messen.

Im Wesentlichen stellen diese ausgewählten Indikatoren Informationen bereit, damit Sie potenzielle Probleme in den Hauptkomponenten des Systems (CPU, Arbeitsspeicher und Speicher) erkennen können. Ein Beispiel für die

Verwendung dieser Leistungsindikatoren in der ColdStorage-Beispielanwendung finden Sie in der MSDN Library:

<http://msdn.microsoft.com/library/en-us/dncold/html/storcpcptyplan.asp> (englischsprachig).

Verteilung der Arbeitsauslastung und ASP.NET-Anwendungen

Wenn Sie eine große Anzahl an Seitenanforderungen für die ASP.NET-Anwendungen erwarten, sollten Sie die Installation einer Webfarm von Webservern mit dem Windows-Netzwerklastenausgleich (Windows Network Load Balancing oder WNLB) in Erwägung ziehen. In diesem Fall können Sie einen virtuellen Zugriffspunkt für die Benutzer bereitstellen, und der WNLB verteilt die Arbeitsauslastung transparent auf alle Mitglieder der Webfarm.

ASP.NET ist im Hinblick auf das Speichern des Sitzungsstatus äußerst vielseitig, wodurch die meisten Probleme beim Ausführen von ASP-Legacyanwendungen in einer Webfarm behoben werden. Sie können die Datei **web.config** einer bestimmten Anwendung verwenden, um die Art der Statusspeicherung anzugeben. Folgende Optionen sind verfügbar:

- **InProc.** In diesem Modus werden die Werte im Arbeitsspeicher des ASP.NET-Arbeitsprozesses gespeichert. Da hier keine prozessübergreifenden Vorgänge beteiligt sind, bietet dieser Modus die schnellste Methode für die Statusspeicherung in ASP.NET. Wenn jedoch aus bestimmten Gründen der Arbeitsthread erneut initialisiert wird oder die Benutzersitzung einen anderen Server in der Webfarm aufruft, gehen sämtliche Statusinformationen vollständig verloren. Anhand des folgenden Beispiels wird veranschaulicht, wie Sie diese Option in der Datei **web.config** festlegen können.

```
<configuration>

  <system.web>

    <sessionState mode="Inproc"

      cookieless="true"

      timeout="20" />

  </sessionState>

</system.web>

</configuration>
```

- **StateServer.** Dieser Modus verwendet einen unabhängigen Windows-Dienst (ASP.NET-Statusdienst oder **aspnet_state**), um Informationen zum Sitzungsstatus zu verwalten. Wenn eine ASP.NET-Anwendung den zugehörigen Arbeitsprozess in einer bestimmten Sitzung wieder verwertet, ist der Status sicher, weil er nicht im lokalen Arbeitsspeicher gespeichert wird, der dem Arbeitsprozess zugeordnet ist. Zum Verbessern der Skalierbarkeit können Sie den Sitzungsstatus auf einem anderen Server speichern. Der ASP.NET-Statusdienst wird daraufhin diese Informationen speichern. Zu diesem Zweck müssen Sie nur den Namen des Servers und den von diesem Dienst verwendeten IP-Port (standardmäßig **42424**) kennen. Ein weiterer Vorteil besteht darin, dass der Sitzungsstatus beim Neustart von IIS nicht verloren geht. Diese prozessübergreifende Kommunikation stellt jedoch eine zusätzliche Belastung dar und ist somit nicht so effizient wie **InProc**. Wenn Sie über eine Webfarm verfügen, stellt das Speichern des Sitzungsstatus auf einem anderen Server in der Regel die beste Lösung dar.

```
<configuration>

  <system.web>

    <sessionState mode="StateServer"

      stateConnectionString="tcpip=stateserver.mydomain.com:42424"

      cookieless="false"

      timeout="20" />

    </sessionState>

  </system.web>

</configuration>
```

Sie können den Standardport für den Dienst **aspnet_state** unter folgendem Registrierungsschlüssel ändern:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aspnet_state\Parameters\Port.

Sie können den Dienst **aspnet_service** in der Systemsteuerung mithilfe des Applets **Dienste** starten und beenden. Sie können auch eine Eingabeaufforderung öffnen und den folgenden Befehl ausführen:

Net start aspnet_state

Net stop aspnet_state.

- **SqlServer. StateServer** ermöglicht Ihnen ggf. eine zentrale Speicherung des Sitzungsstatus in einer Webfarm. Der Dienst **aspnet_service** ist jedoch keine relationale Datenbank und verfügt nicht über die Transaktionsfunktionen, die eine entsprechende Zuverlässigkeit für die meisten relationalen Datenbanksysteme bieten. Sie können den SQL Server zum Speichern von Sitzungen verwenden. Dieser Prozess ist nicht so effizient wie **InProc**, weil die Informationen in einem anderen Prozess oder auf einem anderen Server gespeichert werden müssen. Dieser Prozess könnte jedoch sinnvoller als die Verwendung des Dienstes **aspnet_service** sein. Dies richtet sich nach der tatsächlichen Arbeitsauslastung und nach der Datenbankkonfiguration. Sie können diese Option in der Datei **web.config** wie folgt angeben:

```
<configuration>

  <system.web>

    <sessionState mode="SqlServer"

      sqlConnectionString="data source= mysql.mydomain.com;User
      id=StateUser;password=StatePassword;"

      cookieless="false"

      timeout="20" />

    </sessionState>

  </system.web>

</configuration>
```

Bedenken Sie, dass Benutzer Dateien mit der Bezeichnung **web.config** standardmäßig nicht durchsuchen können. Daher sind sie auch nicht in der Lage, das in dieser Datei gespeicherte Kennwort zu lesen. Sie sollten jedoch trotzdem sicherstellen, dass der entsprechende Benutzername und das Kennwort nur auf die **ASPState**-Datenbank Zugriff haben, in der der Sitzungsstatus gespeichert werden muss. Andernfalls ist ein potenzielles Sicherheitsrisiko gegeben.

Sie können die **ASPState**-Datenbank mithilfe des Skripts **InstallSqlState.sql** entwickeln, das in den .NET Framework-Dateien unter folgender Adresse gespeichert ist:

C:\Windows\Microsoft.NET\Framework\v1.0.3705

(Anmerkung: Sie sollten den korrekten Pfad im jeweiligen Fall überprüfen, weil Windows in diesem Beispiel im Verzeichnis **C:\Windows** installiert ist und es sich um die .NET Framework-Version 1.0.3705 handelt.)

Zwecks erhöhter Zuverlässigkeit können Sie SQL Server in einem Failovercluster einrichten und den Sitzungsstatus vor potenziellen Ausfällen eines bestehenden SQL Server-Dienstes schützen.

Zusätzliche Informationen zur Verteilung der Arbeitsauslastung von ASP.NET-Anwendungen finden Sie im Whitepaper "Performance Tuning a .NET Deployment" (englischsprachig).

Problembehandlung bei ASP.NET-Anwendungen

Mithilfe der Ereignisanzeige können Sie nach Ereignissen suchen, die von ASP.NET-Anwendungen hervorgerufen werden. Sie können auch einen Filter angeben, um nur die ASP.NET-Ereignisse im Anwendungsprotokoll anzuzeigen (siehe Abbildung 13):

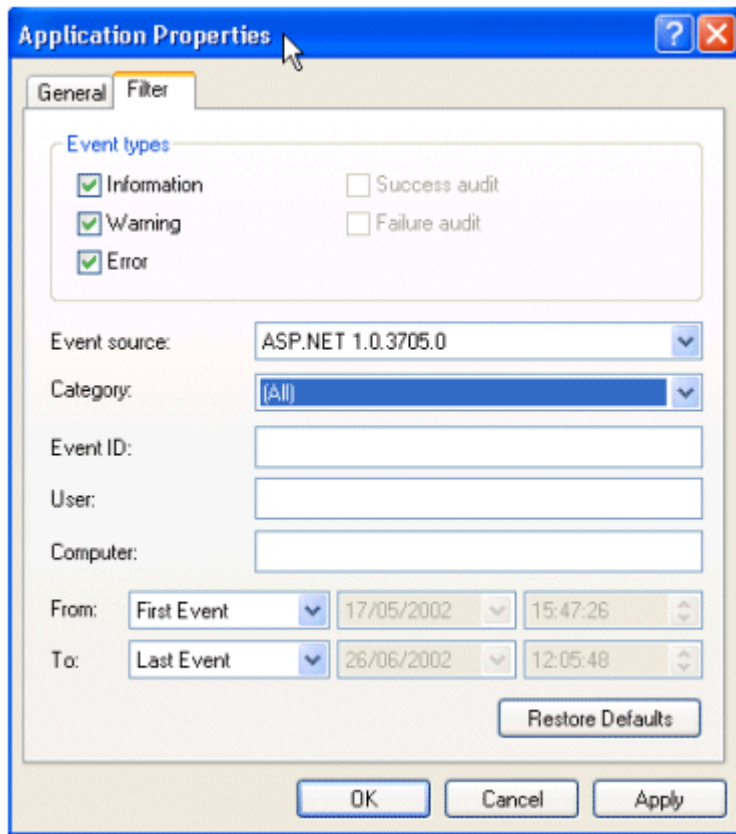


Abbildung 13 Erstellen eines Filters in der Ereignisanzeige, um nur ASP.NET-Ereignisse anzuzeigen

Die bereitgestellten Informationen sind mit den in Abbildung 14 dargestellten Informationen vergleichbar.

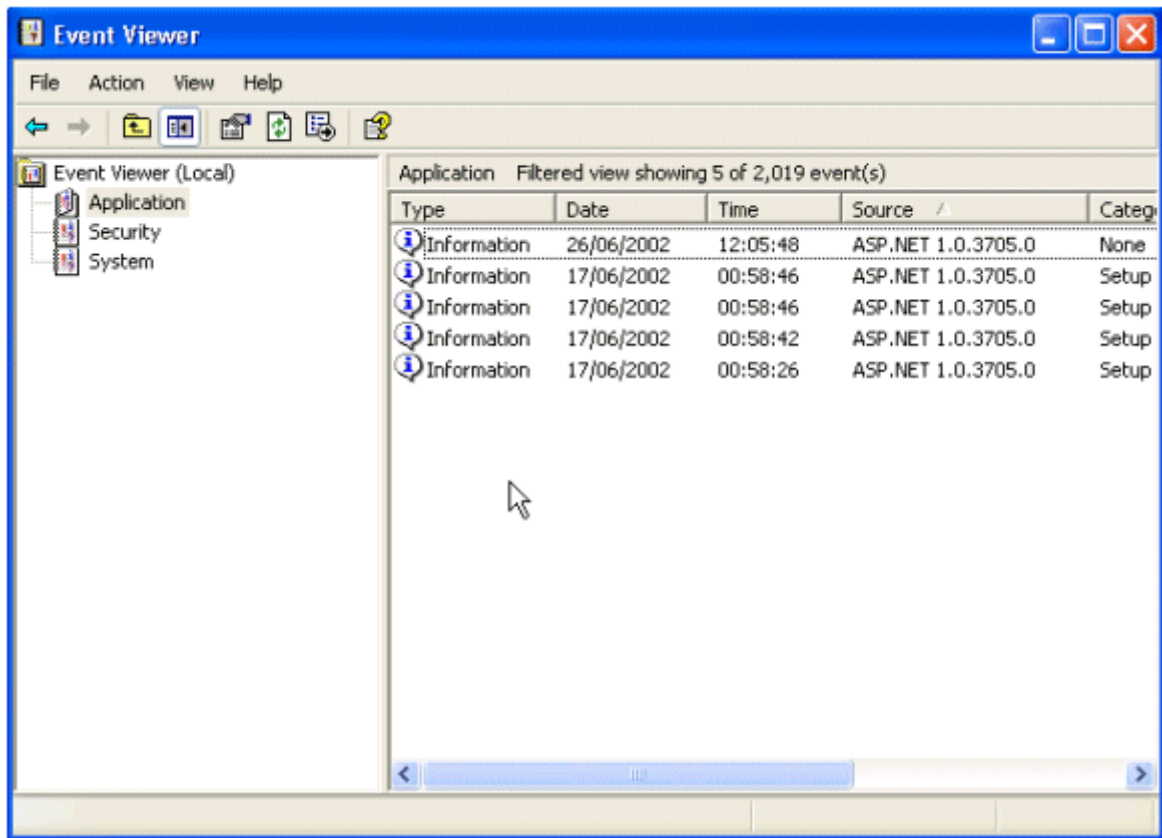


Abbildung 14 Anzeigen von gefilterten ASP.NET-Ereignissen in der Ereignisanzeige

Ablaufverfolgung von ASP.NET-Anwendungen

ASP.NET stellt ein leistungsstarkes Tool zum Debuggen von Anwendungen durch die Ablaufverfolgung von Debuganweisungen bereit. Informationen zu diesem Feature finden Sie auf der folgenden Website:

<http://msdn.microsoft.com/library/en-us/cpguide/html/cpcontracefunctionality.asp> (englischsprachig).

Bei Produktionsservern sollte diese Funktion jedoch deaktiviert sein. Sie erreichen dies, indem Sie den Abschnitt `<system.web>` der Datei **web.config** für die entsprechende ASP.NET-Anwendung bearbeiten und sicherstellen, dass die Ablaufverfolgung wie in der folgenden Konfigurationsdatei deaktiviert ist:

```
<configuration>

  <system.web>

    <trace enabled="false"

      pageOutput="true"

      requestLimit="15" />

  </system.web>

</configuration>
```

Zusätzliche Informationen zur Ablaufverfolgung von ASP.NET-Anwendungen finden Sie im Whitepaper "Performance Tuning a .NET Deployment Guide" (englischsprachig).

Verwalten von .NET-Webdiensten

Neben den im Zusammenhang mit ASP.NET-Anwendungen erläuterten Informationen, die in den meisten Fällen auch für Webdienste gelten, stellt das .NET Framework Software Development Kit Tools zum Entwickeln, Bereitstellen und Verwalten von Anwendungen und Komponenten auf der Grundlage des .NET Framework bereit. Von einer Beschreibung der einzelnen Tools wird in diesem Dokument abgesehen. Eine vollständige Beschreibung der Tools finden Sie jedoch auf der .NET Framework SDK-Website unter:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpconnetframeworktools.asp> (englischsprachig).

Überwachen von und Problembehandlung bei .NET-Webdiensten

Mithilfe des Systemmonitors erhalten Sie Zugriff auf bestimmte Indikatoren (z.B. die CPU- und Arbeitsspeichernutzung) für einen speziellen Webdienst. In einigen Fällen können Sie die Aktivitäten von Webdiensten indirekt überwachen, indem Sie beispielsweise auf den Indikator **Internetinformationsdienste Bereitstellungsanforderungen/s** achten, um die Anforderungen an den virtuellen Server zu verfolgen. Wenn dieser Server hauptsächlich als Server der mittleren Ebene verwendet wird, auf dem Webdienste ausgeführt werden, gibt dieser Indikator Aufschluss über die Auslastung der Webdienste.

Wenn Sie zum Senden von Anforderungen an Webdienste HTTP einsetzen, können Sie die im Lieferumfang der im IIS enthaltenen Protokollierungsfunktionen zum Überwachen dieser Anforderungen über IIS verwenden. Mithilfe der Ereignisanzeige können Sie das Ereignisprotokoll für einen bestimmten Webdienst anzeigen. Und wenn dieser Webdienst den Namespace **System.Diagnostics.EventLog** optimal verwendet, können diese Informationen bei der Problembehandlung bei Webdiensten und .NET-Anwendungen im Allgemeinen extrem hilfreich sein. Sie können beispielsweise die **EventLog**-Klasse nutzen, um in dem Ereignisprotokoll nach bestimmten Ereignissen zu suchen und neue Ereignisse in das Ereignisprotokoll zu schreiben. Dadurch stehen umfassendere Informationen für die Problembehandlung der Anwendung als die vom .NET Framework und anderen Windows-Diensten produzierten Standardfehlerinformationen zur Verfügung.

Das Debuggen einer Anwendung kann eine sinnvolle Technik der Problembehandlung darstellen. Sie sollten jedoch die Webdienste in einer Testumgebung debuggen, da das Produktionssystem ansonsten störenden – wenn auch nur periodisch auftretenden – Ausfallzeiten ausgesetzt ist. In einigen Fällen können Sie Anwendungen mithilfe von Ablaufverfolgungsmethoden debuggen. Auch dies sollte nur in Testumgebung erfolgen, um einen unnötigen Overhead des Produktionsservers zu vermeiden. Zusätzliche Informationen zur Problembehandlung bei Webdiensten finden Sie unter:

<http://msdn.microsoft.com/library/en-us/dnservice/html/service12192001.asp> (englischsprachig).

Verteilte Transaktionsverwaltung

Wenn die Webdienste Transaktionsfunktionen von COM+ verwenden, wie beispielsweise bei automatischen Transaktionen, können Sie mithilfe der Komponentendienste (**C:\WINDOWS\system32\Com\comexp.msc**) auf Informationen zu diesen Transaktionen zugreifen. Dieses Tool ist über das Startmenü als Teil der Programmgruppe **Verwaltung** verfügbar (siehe Abbildung 15).

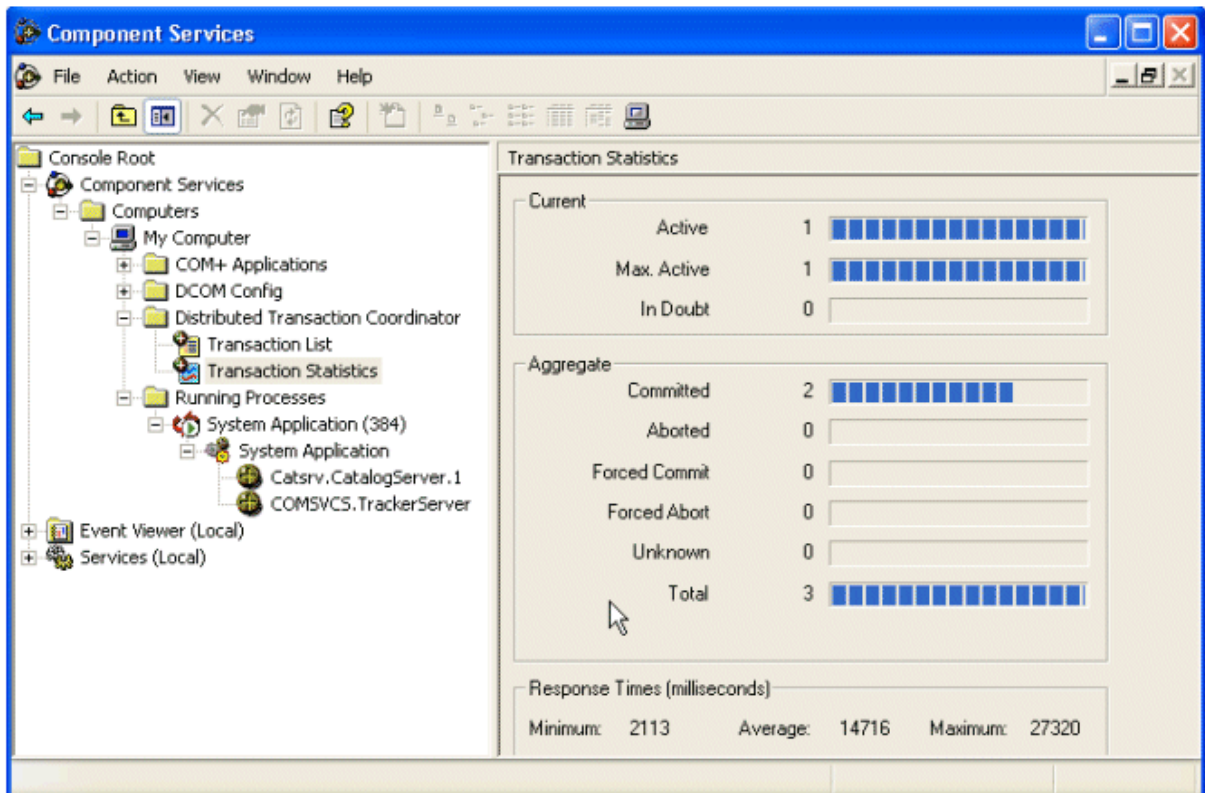


Abbildung 15 Anzeigen der verteilten Transaktionsstatistik mit Komponentendiensten

Dieses Tool bietet Ihnen nicht nur Informationen zur Anzahl der gleichzeitig auf dem System ausgeführten Transaktionen, sondern auch zu den Antwortzeiten, so dass Sie das Transaktionssystem optimieren können.

Das Microsoft .NET Framework stellt eine leistungsstarke und stabile Plattform bereit, auf der sichere und skalierbare Server- und Clientanwendungen ausgeführt werden können. Viele wichtige Funktionen von .NET-Lösungen basieren auf Netzwerkinfrastruktur- und Back-End-Diensten, wie IIS, SQL Server und andere Produkte des .NET Enterprise Servers. Dieses Dokument enthält detaillierte Anweisungen zum Ausführen und Verwalten von .NET-Anwendungen und den zugehörigen IT-Systemen. Es ist jedoch wichtig, die Bedeutung angemessener und konsistenter Verwaltungsstrukturen zu kennen, wie sie beispielsweise vom MOF vorgeschlagen werden, um eine IT-Infrastruktur, die auf NET-Anwendungen basiert, erfolgreich verwalten zu können.