

Kapitel 1

Willkommen zu Windows 7

In diesem Kapitel:

Was hat sich seit Windows Vista geändert?	20
Sieben Wege zu einem glanzvollen Windows 7	21
Zusammenfassung	36

Willkommen zu Windows 7, dem neuesten Betriebssystem von Microsoft, das sich durch großartige Technik, viele Innovationen, interessante neue Features und Funktionalität auszeichnet, die die Benutzerfreundlichkeit erheblich verbessern!

Windows 7 ist die bei weitem schnellste, am einfachsten zu verwendende und faszinierendste Version von Windows, da sie auf eine neue Hardwaregeneration ausgerichtet ist – zum Beispiel Bildschirme oder Geräte, die Touch-Eingaben (auch mit mehreren Touch-Punkten) unterstützen, Umgebungssensoren, Standortanbieter und mehr. Außerdem bietet das Betriebssystem einige großartige neue Dienste, wie zum Beispiel eine integrierte Multifunktionsleiste in der Art der Microsoft Office-Anwendungen und die Windows 7-Taskleiste. Somit steht eine Plattform zur Verfügung, die ihren Benutzern erlaubt, mit ihrer Anwendung einfacher zu interagieren.

Ein weiteres neues Feature ist die Windows-Problembehandlungsplattform, die einheitliche automatisierte Fehlerbehebungsfunktionen realisiert. Diese Plattform erlaubt es, Probleme in Windows und in ihrer Anwendung nach einem Standardverfahren zu erkennen, aufzulösen und zu überprüfen – die Benutzerzufriedenheit verbessert sich.

Über diese entwicklerorientierten Features hinaus ist Windows 7 generell ein Upgrade mit einem neuen Suchsystem, das es Ihnen erlaubt, bei den immer größer werdenden Festplatten und den immer umfangreicheren Datenmengen, die Computer speichern, die gewünschten Elemente schneller und gezielter zu finden. Das Windows Media Center als Zentrum für das digitale Heim wurde mit einer neuen Benutzeroberfläche und Unterstützung für mehr Hardware und HDTV vollständig überarbeitet.

Schließlich liegt der Fokus dieses Buchs auf einem speziellen Satz von neuen Entwicklerfeatures und Anwendungsprogrammierschnittstellen (APIs), mit denen Sie bessere Anwendungen erstellen können, die von den neuen Features in Windows 7 profitieren. Beachten Sie, dass in diesem Buch nur die Dinge betrachtet werden, die für Entwickler neu sind. Wenn Sie also noch nie eine Windows-Anwendung erstellt haben, ist es durchaus möglich, dass Sie einigen Themen im Buch nicht auf Anhieb folgen können. Allerdings haben wir versucht, Ihnen mit den Beispielen vollständige Anwendungen zu zeigen, die auf die neuen Features zurückgreifen.

Was hat sich seit Windows Vista geändert?

Möglicherweise haben Sie Ihre Windows-Anwendung bereits auf Windows Vista aktualisiert, vielleicht sind Sie aber auch mit dem zufrieden, was Windows XP zu bieten hat. So oder so ist es verständlich, dass Sie eventuell davor zurückschrecken, sich auf einige durchgreifende Änderungen im Betriebssystem einzustellen.

Als Entwickler wollen Sie eigentlich nur in der Lage sein, Anwendungen zu erstellen, die bestimmten Anforderungen entsprechen, und dabei die Anwendungscompatibilität zu bewahren, damit die Anwendung nicht mit dem neuen Betriebssystem kollidiert. Gleichzeitig möchten Sie aber dem Endanwender eine Arbeitsumgebung bieten, die er versteht und in der er vertraut navigieren kann.

Das Entwicklerteam von Windows 7 bei Microsoft hat sich von diesen Ansprüchen leiten lassen. Obwohl Windows 7 auf dem mit Windows Vista gelegten Fundament aufbaut, hat man die Grundelemente des Betriebssystems verbessert und eine felsenfeste Infrastruktur geschaffen, die Sie in die Lage versetzt, attraktive Benutzerszenarien zu realisieren.

Hunderte von Änderungen wurden an der zugrunde liegenden Infrastruktur in Windows 7 vorgenommen. Wichtig ist, dass dies keine Verhaltensänderungen sind. Es handelt sich vielmehr um Anpassungen des Betriebssystems, um dessen Leistung und Zuverlässigkeit zu verbessern. Windows 7 hat einen wesentlich geringeren Festplattenbedarf als Windows Vista, kommt mit deutlich weniger Arbeitsspeicher aus, verbraucht weniger Prozessorressourcen im Leerlauf, führt weniger E/A-Aktivitäten aus und – alles in allem – ist die Leistungsaufnahme des Systems geringer. Gleichzeitig reagiert Windows 7 flüssiger auf Benutzereingaben und zeichnet sich durch eine bessere Performance aus. Die Benutzeroberfläche macht einen aufgeräumten Eindruck und Benutzerinteraktionen mit dem System – egal, ob es Eingaben behandelt oder das Startmenü oder die Taskleiste anzeigt – finden unverzüglich statt.

Windows 7 wird schneller als je zuvor geladen und lässt sich auf bis zu 256 Prozessoren skalieren. Diese Zahl mag sich riesig anhören, doch sollten Sie bedenken, dass Windows 7 für die nächsten fünf bis acht Jahre seinen Dienst verrichten soll, und es ist mehr als wahrscheinlich, dass in diesem Zeitraum derartige Multicore-Desktopsysteme auf dem Markt sein werden. Änderungen wurden an den internen Threading-Dispatchern vorgenommen, wobei Sperren von kritischen Ressourcen entfernt wurden, um schnellere Kontextwechsel zu ermöglichen. Alles dies (und noch viel mehr) wurde getan, während bahnbrechende Änderungen minimiert wurden und Stabilität, Zuverlässigkeit und Leistung im Mittelpunkt standen. Und vor allem werden Sie nicht von Änderungen belästigt, die Sie zwingen, Zeit und Geld für nicht funktionierende Features aufzuwenden.

Microsoft ist sich auch bewusst, dass ein Windows-Entwickler nur erfolgreich sein kann, wenn das Betriebssystem seine Anstrengungen unterstützt. Deshalb muss es Tools und Techniken integrieren, die dem Entwickler einen Produktivitätsschub garantieren. Zum Lieferumfang von Windows 7 gehört eine große Anzahl von Entwicklungstools und -Frameworks, um Ihnen bei Ihren Entwicklungsanstrengungen unter die Arme zu greifen und die Bereitstellung zu erleichtern. Technologien wie .NET 3.5 Service Pack 1 (SP1), Windows PowerShell 2, MSI 5.0, Native Web Services API und ein neues Menübandframework sind nur einige Beispiele für die in Windows 7 integrierten Technologien.

Dies ist Ihre Gelegenheit, neue und attraktive Benutzerszenarien mit den großartigen Features, die Windows 7 dem Entwickler bietet, zu schaffen.

Sieben Wege zu einem glanzvollen Windows 7

Windows 7 bietet zahlreiche Technologien und Features, um Ihre Anwendung ins rechte Licht zu rücken und sie brillieren zu lassen. In diesem Buch konzentrieren wir uns auf die Features, die unserer Ansicht nach den meisten Zusatznutzen für das Endanwenderszenario bringen und relativ wenig Entwicklungsaufwand verlangen. Unterm Strich bringen Ihnen diese Features für Ihre Investitionen in Zeit, Aufwand und Geld den höchsten Ertrag für die Benutzerfreundlichkeit insgesamt.

Es gibt allerdings kein so herausragendes Feature von Windows 7, das Ihre Anwendung in die beste Windows-Anwendung umwandelt. Die integrierten Windows-Features helfen Ihnen, erstaunliche Anwendungen zu erstellen, doch liegt es bei Ihnen, die richtige Entscheidung zu treffen, welche Features am besten zu Ihren Anwendungsszenarien passen. Dabei können Sie jede Kombination von Windows 7-Features herausgreifen und auswählen. Alles spielt zusammen und Sie können eine solide Anwendung bauen, die die richtigen Features zur richtigen Zeit verwendet. Das beginnt bei nichtfunktionalen Features wie zum Beispiel

Neustart und Wiederherstellung oder Kernel-Transaktions-Manager und setzt sich fort bis zu eher benutzerorientierten Features wie zum Beispiel der neuen Taskleiste, Windows 7-Bibliotheken oder Multitouch. In diesem Buch konzentrieren wir uns auf die wichtigsten Features und diejenigen, die Sie höchstwahrscheinlich einsetzen werden, während Sie Ihre Anwendung mit Windows 7 brillieren lassen:

- Taskleiste
- Bibliotheken
- Multitouch
- Sensoren und Ortung
- Menüband
- Verbesserte Grundelemente
- Erweiterte Grafikplattform

Diese sieben Features können Sie untereinander und mit beliebigen anderen Windows 7-Features kombinieren. Alle Features in der obigen Liste sind voneinander unabhängig. Es gibt kein Feature, das Sie implementieren müssen, doch ist es zweckmäßig, Ihre Anwendung wie eine Windows 7-Anwendung aussehen zu lassen, indem Sie ihr Taskleistenverhalten optimieren und auf die ordnungsgemäße Zusammenarbeit mit Windows 7-Bibliotheken ausrichten. Als Nächstes können Sie damit beginnen, ihre Anwendung von anderen zu differenzieren, indem Sie anspruchsvolle Benutzerszenarien der nächsten Generation schaffen, die Multitouch, Sensoren, Ortung, das neue Menüband und den neuen Grafikstack beinhalten.

Der übrige Teil dieses Kapitels gibt Ihnen einen kurzen Überblick über die verschiedenen Technologien, die im Mittelpunkt dieses Buchs stehen.

Taskleiste

Sie haben nun Windows 7 installiert, melden sich das erste Mal an und warten begierig darauf, dass die Aufforderung »Desktop wird vorbereitet ...« verschwindet. Die erste Änderung der Benutzeroberfläche, die Sie bei Windows 7 bemerken, ist die neue überarbeitete Windows-Taskleiste.

Im Lauf der Jahre hat sich die Windows-Taskleiste zu der überladenen Form entwickelt, mit der Sie vielleicht von Windows Vista her vertraut sind. Die Reichhaltigkeit von Startoberflächen hat es Benutzern erschwert, zu ihren bevorzugten Programmen zu gelangen, zwischen Anwendungen zu wechseln und auf häufig benötigte Ziele zuzugreifen. Die neue Taskleiste spiegelt viele Jahre Benutzeroberflächendesign und Benutzerfreundlichkeitstests wider und verkörpert eine Revolution im Design von Startoberflächen. Nie zuvor ist es so einfach gewesen, in Windows und Fenstern zu navigieren – die neue Benutzeroberfläche ist einfach, intuitiv, frisch und sauber. Benutzern wird es leichter fallen, ihre bevorzugten Anwendungen zu starten, zwischen Fenstern zu wechseln, Anwendungen zu steuern, die den Fokus momentan nicht besitzen, und zu ihren letzten Dokumenten zu gelangen – all dies ist nur einen Mausklick entfernt. Abbildung 1.1 zeigt den schlanken, sauberen Look der Windows 7-Taskleiste und des Startmenüs.

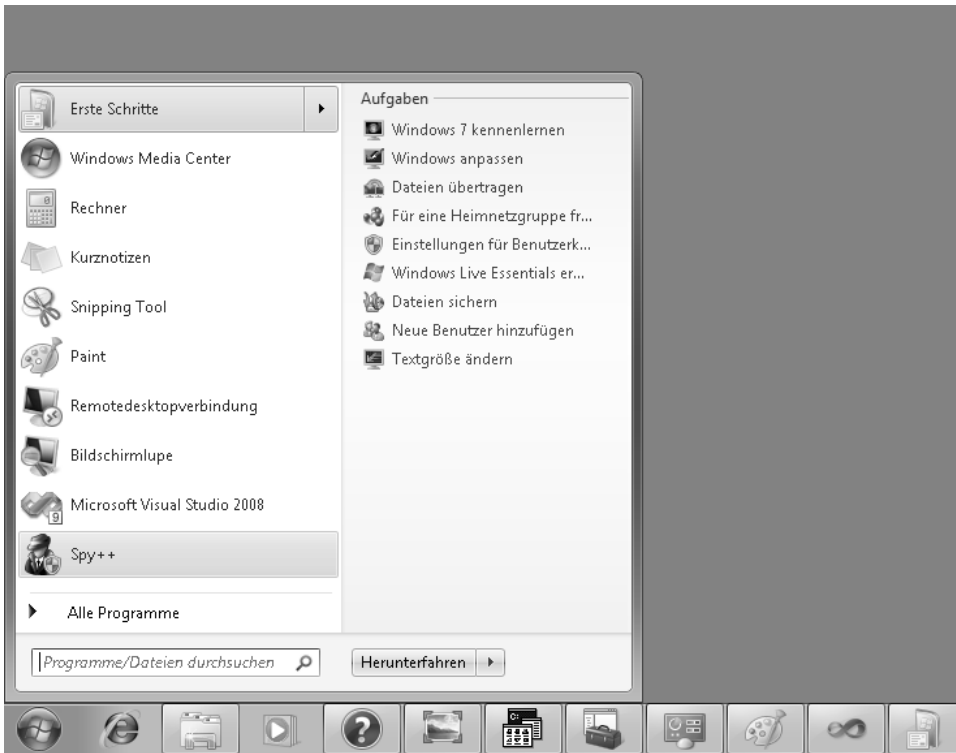


Abbildung 1.1 Die Windows 7-Taskleiste und das Startmenü

Dieses Buch stellt die neuen Features der Windows 7-Taskleiste vor und erläutert, wie Sie gewährleisten, dass Ihre Anwendung von jedem relevanten Teil der Funktionalität profitieren kann. In Kapitel 2 erfahren Sie, wie Sie die neuen Features der Taskleiste verwenden, die relativ wenig Aufwand von Ihrer Seite aus verlangen – Bereitstellen einer Fortschrittsanzeige über eine Taskleistenschaltfläche, Anzeigen eines Overlaysymbols, das zusätzliche Statusinformationen liefert, und Differenzieren von Fenstern und Anwendungen, indem ihnen getrennte Taskleistenschaltflächen zugewiesen werden. In Kapitel 3 geht es dann um die ausgefeilteren Features, die Benutzer möglicherweise von ihrer Anwendung erwarten – Miniaturansicht-Symbolleisten für die Remotesteuerung eines Fensters von seiner Taskleiste-Miniaturansicht, Sprunglisten für schnellen Zugriff auf die letzten Dokumente und häufige Aufgaben, Tools für die vollständige Anpassung von Taskleiste-Miniaturansichten und Tools für das Erstellen von Live-Fenstervorschauen. Alle diese Features lassen sich kombinieren, um eine attraktive Funktionalität für Benutzer Ihrer Anwendungen bereitzustellen.

In diesen beiden Kapiteln erfahren Sie, wie Internet Explorer Live-Vorschauen seiner internen Webseitenregister darstellt, wie Sie mit Windows Media Player von der Miniaturansicht der Anwendung Musik wiedergeben können, wie Windows Live Messenger es Ihnen erlaubt, Ihre Online-Verfügbarkeit zu ändern und zu Ihrem E-Mail-Posteingang von seiner Sprungliste zu navigieren, wie das Feature *Kurznotizen* Ihnen eine Miniaturansicht eines Stapels von Notizen zeigt und Ihnen erlaubt, mit einem einzigen Mausklick eine andere Notiz zu erzeugen. Abbildung 1.2 zeigt Internet Explorer 8-Registerkarten als Miniaturansichten von Taskleistenschaltflächen.



Abbildung 1.2 Miniaturansichten von Symbolleistenschaltflächen

In den genannten Kapiteln sehen wir uns die systemeigenen Windows-APIs an, die diese Wunder möglich machen, einschließlich der zugrundeliegenden COM-Schnittstellen, Fensternachrichten und Win32-Funktionen. Außerdem verwenden wir das Windows API Code Pack für Microsoft .NET Framework, um mit Windows 7 von verwaltetem Code aus zu interagieren. Wir tun dies mithilfe einer gut konzipierten Abstraktionsschicht für die Organisation von Taskleistenfeatures, die Ihnen sofort vertraut sein dürften, wenn Sie mit dem .NET Framework Code schreiben.

Nahezu jede Anwendung benötigt die neue Taskleiste, um wirklich mit Windows 7 zu glänzen. Wenn Sie eine dokumentorientierte Anwendung schreiben, können Sie Ihren Benutzern unmittelbaren Zugriff auf ihre zuletzt und häufig verwendeten Dokumente geben. Wenn Sie eine Anwendung mit umfangreichen Medien entwickeln, können Sie Ihren Benutzern eine Miniaturansicht-Symbolleiste für schnellen Zugriff auf medienorientierte Operationen und lebendige Live-Vorschauen Ihrer Anwendung von der Taskleistenschaltfläche aus bieten. Wenn Sie ein Hilfsprogramm programmieren, das im Hintergrund werkelt und sich nur hin und wieder zeigt, werden Sie feststellen, dass Ihre Anwendung am besten in der Windows-Taskleiste aufgehoben ist, wo Sie Benutzern eine Live-Fortschrittsanzeige und ein Statussymbol bieten können. Selbst wenn Sie für den Browser programmieren, werden Sie eines Tages feststellen, dass Ihre Benutzer versessen darauf sind, Ihre Anwendung außerhalb des Browsers auszuführen, weil ihnen die Aufmachung der Windows 7-Taskleiste gefällt.

Bibliotheken

Seit Windows XP hatten Sie die Option, Ihre Inhalte (Dokumente, Videos, Bilder oder beliebige andere Dateien) in einem Satz speziell dedizierter Ordner wie zum Beispiel *Eigene Dokumente* und *Eigene Bilder* zu speichern. Diese Ordner sind für jeden Benutzer eindeutig und in Windows Vista wurden sie automatisch indiziert, um eine überragende Suchleistung zu bieten. Doch haben Sie es sich nicht gewünscht, diese speziellen Ordner erweitern und zusätzliche Ordner auf Ihrer lokalen Festplatte oder auf Remotefreigaben hinzuzufügen zu können? Bibliotheken in Windows 7 erlauben Ihnen, genau das zu tun!

Mit Windows 7-Bibliotheken kontrollieren Sie Ihre Daten. Windows 7-Bibliotheken erlauben es, Ihren eigenen Satz von Ordnern für verschiedene Zwecke zu verwalten oder weitere Ordner zur Bibliothek *Eigene Bilder* hinzuzufügen. Bibliotheken sind benutzerdefinierte Sammlungen von Ordnern, die zu einer vereinheitlichten Ansicht aggregiert werden. Durch das Einbinden von Ordnern in Bibliotheken sagt der Benutzer Windows (und Ihnen, dem Entwickler), wo seine wichtigen Daten untergebracht sind.

Windows 7-Bibliotheken sind der neue Einstiegspunkt der Benutzer für den Zugriff auf ihre Daten. Benutzer gehen zu Bibliotheken, um ihre Dokumente, Musik, Bilder und andere Dateien zu verwalten. Und Benutzer erwarten, dass Ihre Anwendung mit Bibliotheken zusammenarbeitet, weil sie einen integralen Teil der Windows-Shell ausmachen und eine dominante Rolle innerhalb von Windows-Explorer und anderen Teilen des Betriebssystems spielen, beispielsweise dem allgemeinen Dateidialogfeld.

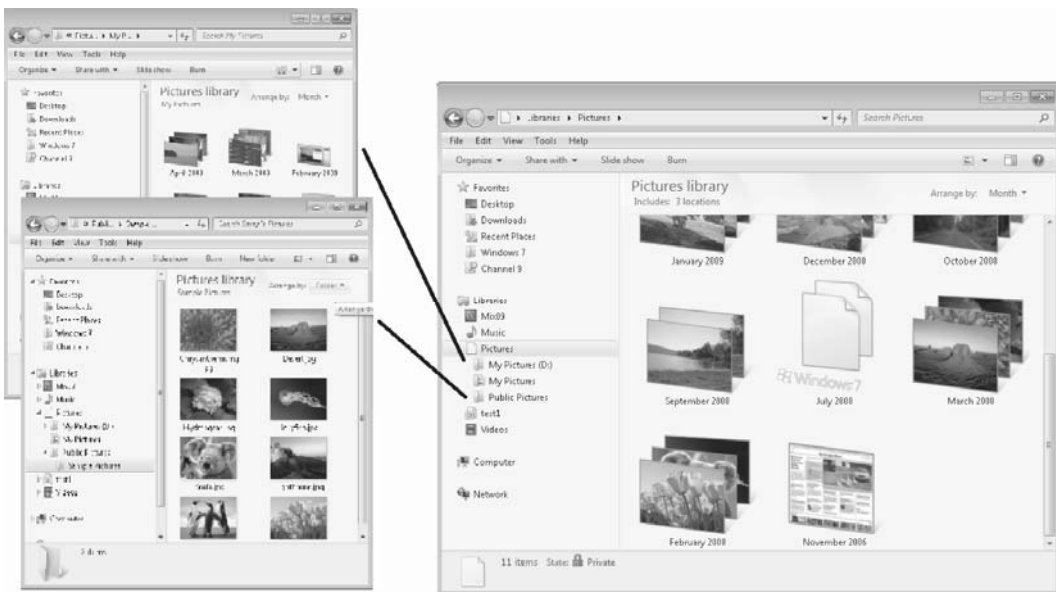


Abbildung 1.3 Benutzeroberfläche mit Bibliotheken

Eine Bibliothek ähnelt in gewisser Weise einem Ordner. Vom Standpunkt eines Benutzers aus sieht eine Bibliothek wie ein normaler Ordner aus und verhält sich auch so. Jedoch fasst eine Bibliothek im Gegensatz zu einem Ordner Dateien, die an verschiedenen Orten abgelegt sind, in einer vereinheitlichten Ansicht zusammen. Dieser Unterschied zwischen einer Bibliothek und einem Ordner ist fein, aber entscheidend. Bibliotheken speichern tatsächlich keine Elemente eines Benutzers, sie überwachen lediglich Ordner, die

Elemente enthalten, und bieten dem Benutzer die Möglichkeit, auf diese Elemente zuzugreifen und sie in verschiedener Weise zu arrangieren. Führen Sie ein kleines Experiment durch. Starten Sie den Editor (*Noteepad.exe*) und speichern Sie die Datei in der Bibliothek *Dokumente*. Die Speichern-Operation lässt sich erfolgreich abschließen. Allerdings haben wir eben festgestellt, dass Bibliotheken keine normalen Ordner sind, die Dateien enthalten. Was also passiert hier?

In Kapitel 4 erfahren Sie mehr zu Windows 7-Bibliotheken. Dort geht es um die zugrundeliegende Architektur des Features *Bibliotheken* und es wird gezeigt, warum das Speichern von Dateien in der Bibliothek *Dokumente* (oder jeder anderen Bibliothek) in Windows 7 funktioniert. Dieses Kapitel erläutert auch im Detail die API, die erforderlich ist, um mit Windows 7-Bibliotheken zu arbeiten und diese zu verwalten. Möglicherweise unterstützt Ihre Anwendung bereits Bibliotheken, doch selbst wenn das nicht der Fall ist, werden Sie sehen, dass kein großer Aufwand erforderlich ist, eine Anwendung in eine bibliothekenfähige Anwendung umzuwandeln, die mit Windows 7-Bibliotheksinhalten arbeiten kann. Wie in den meisten Kapiteln dieses Buchs stellen wir sowohl die nativen C++ (vorwiegend COM)-APIs und – wo es möglich ist – die APIs für verwalteten Code vor. Wir erläutern die Architektur und die Konzepte, die auf die systemeigene API zurückgreifen, und tauchen in spezifische Beispiele ein, wie die neue *Libraries*-API, mithilfe der API für verwalteten Code eingesetzt wird.

Fast jede Windows-Anwendung, die es Benutzern erlaubt, Dateien auf der lokalen Festplatte zu speichern und von dort zu lesen, wird die Windows 7-Bibliotheken nahtlos integrieren, damit Benutzer in den Genuss der umfangreichen Such- und Anzeigemöglichkeiten kommen, die Windows Explorer bietet, und die Arbeit in der Windows 7-Umgebung ohne Schwierigkeiten vonstatten geht.

Touch, Multitouch und Bewegungen

Bei Demonstrationen von Windows 7 werden Touch, Multitouch und Bewegungen wahrscheinlich als herausragende Features vorgeführt. Da Microsoft zuerst die Plattformoberfläche demonstriert hat, ist das Interesse an Anwendungen und Szenarien, die dem Benutzer erlauben, den Bildschirm an mehreren Stellen zu berühren, um mit Benutzeroberflächen zu interagieren, stark gewachsen! Windows 7 unterstützt Touch-Oberflächen vollständig und ist ausreichend flexibel, um viele unterschiedliche Touch-Typen zu behandeln. Vielleicht sind Sie mit dem Erstellen von Anwendungen für Touch-Eingaben mit Tablet PCs vertraut, die das Freihand-Subsystem verwenden, doch das ist nur der Anfang.

Aufgrund der Bedeutung und Popularität dieser Features räumen wir dem Thema Multitouch-APIs einen breiten Raum ein. Zuerst zeigen wir im Detail, wie das zugrundeliegende System funktioniert und wie das Betriebssystem die unterschiedliche Touch-Hardware verwaltet. In der Vergangenheit war Touch-Hardware gleichbedeutend mit einem Tablet PC, der lediglich die Eingabe über Stift erlaubte, oder mit spezialisierter Hardware, die spezielle Treiber verlangte, in die sich ein Entwickler erst intensiv einarbeiten musste. Mit Windows 7 können Hardwareanbieter jetzt Bildschirme oder externe Geräte bauen und sie in das neue Touch-Treibersystem einklinken – und Sie als Entwickler profitieren unmittelbar davon.

In Kapitel 5 behandeln wir die grundlegenden Konzepte der Windows-Multitouch-Programmierung, stellen die Architektur im Detail vor und untersuchen eingehend die nativen C++ (COM)-APIs. Dieses Kapitel konzentriert sich hauptsächlich auf das in Windows 7 integrierte Bewegungsmodul, das Entwickler in die Lage versetzt, ihre Anwendung mit sehr geringem Aufwand mit Multitouch-Unterstützung auszustatten.

In Kapitel 6 werfen wir einen Blick auf das .NET Framework und insbesondere die Windows Presentation Foundation (WPF). Das .NET Framework verkörpert einen Satz von Klassen, die Microsoft für Sie erstellt hat, um eine komfortable Programmierschnittstelle bereitzustellen, die sich mit Sprachen wie zum Beispiel Microsoft Visual Basic oder C# verwenden lässt. Ein Teil des .NET Framework ist die Windows Presentation Foundation, die dafür konzipiert ist, Zugriff auf alles zu bieten, was Sie brauchen, um die visuelle Schicht für die reichhaltige Benutzeroberfläche einer Anwendung zu erstellen. Natürlich fällt das Touch-System unter diese Domäne und Version 4.0 von WPF bietet nicht nur Zugriff auf das Touch-System, sondern auch auf eine Gruppe von APIs, die Sie über Benutzerbewegungen informieren.

Eine Bewegung (*Gesture*) ist eine festgelegte Aktion, die in einer Multitouch-Umgebung auftreten kann. Stellen Sie sich dazu vor, wie Sie mit einer Fotografie in einer Multitouch-Umgebung interagieren. Normalerweise wird eine Fotografie als Rechteck ausgegeben, das vertikal ausgerichtet ist. Gewöhnlich können Sie es mit einem Mausrad oder einem Schieber vergrößern und verkleinern. Wie wäre es, wenn Sie den Bildschirm berühren könnten? Wäre es nicht praktisch, wenn Sie den Daumen auf eine Ecke legen und Ihren Zeigefinger auf die entgegengesetzte Ecke und dann Ihre Finger zusammenführen, um das Bild kleiner zu machen, da die Ecken Ihren Fingern folgen? Auf ähnliche Art und Weise können Sie das Bild vergrößern, indem Sie Ihre Finger auseinanderziehen, das Bild drehen, indem Sie Ihre Handgelenke drehen, oder es an einen anderen Platz auf dem Bildschirm verschieben, indem Sie Ihre Hand bewegen und das Bild ihr folgen lassen. Dies sind die Konzepte, die in Windows 7 als *Bewegungen* bezeichnet werden. In Kapitel 6 erfahren Sie, wie Sie Ihre Anwendung ganz leicht mit Bewegungsunterstützung ausstatten können.

In Kapitel 7 vertiefen wir unsere Diskussion und tauchen in leistungsfähigere und komplexere Multitouch-APIs ein. Dieses Kapitel behandelt Windows 7-Raw-Touch-Unterstützung und die systemeigenen Manipulations- und Trägheitsprozessoren.

In diesen beiden Kapiteln unternehmen Sie eine vollständige Tour durch die Multitouch-Features. Am Ende verfügen Sie über profunde Kenntnisse, wie das Touch-System in Windows 7 arbeitet. Diese beiden Kapitel sind zwar für die beiden folgenden Kapitel, die sich mit der Windows Presentation Foundation und Silverlight-Multitouch beschäftigen, nicht unbedingt erforderlich, versorgen Sie aber mit Informationen über die relevanten Teile der Architektur, sodass es Ihnen leichter fällt, die .NET-Multitouch-Abstraktionsschichten zu verstehen.

Im Jahre 2007 hat Microsoft die Technologie Silverlight veröffentlicht. Silverlight ist ein Browser-Plug-In, das eine umfangreiche Programmierschnittstelle und eine Miniaturversion des .NET Framework bietet, mit der sich Anwendungen, die von den Features des Betriebssystems profitieren können, erstellen und leicht bereitstellen lassen. Ein attraktives Feature von Silverlight ist die Möglichkeit, die Anwendung außerhalb des Browsers auszuführen, wobei ein Benutzer wählen kann, die Abhängigkeit der Anwendung vom Browser zu entfernen und sie auf seinem Desktop zu installieren. In Kapitel 8 werfen wir einen Blick auf Silverlight – was es ist und was es tut. Dort wird gezeigt, wie leicht es ist, eine Anwendung in .NET zu erstellen, die sich mithilfe von Silverlight über das Internet bereitstellen lässt. Dann erfahren Sie, wie Sie Touch-Eingaben zu Ihrer Silverlight-Anwendung in Windows 7 mithilfe der Freihandfunktionen für Tablet PCs sowie den Raw-Touch-Objekten, mit denen sich Multitouch-Nachrichten verarbeiten lassen, hinzufügen.

Am Ende von Kapitel 8 wissen Sie alles über Touch-Eingaben und eine ganze Menge mehr – vom zugrunde liegenden Touch-System über die .NET-APIs für Bewegungen, die durch WPF bereitgestellt werden, bis hin zum Hinzufügen von Touch-Funktionalität zu reichhaltigen Webanwendungen.

Sensoren und Ortung

Die Plattform für Windows-Sensoren und -Ortung gehört zu den attraktivsten und nützlichsten neuen Technologien von Windows 7 – und zwar sowohl für Benutzer als auch für Entwickler. Damit kann der Entwickler adaptive Anwendungen erstellen, die ihr Aussehen und Verhalten an unterschiedliche Umgebungsbedingungen anpassen. Doch weshalb sollten Sie sich überhaupt mit derartiger Funktionalität befassen? Sehen Sie sich um. Es gibt mobile Geräte, die ihre Anzeigerausrichtung nach der relativen Position des Geräts ändern. Immer mehr Computer enthalten Umgebungslichtsensoren, die es erlauben, die Anzeigehelligkeit des Bildschirms an die aktuellen Beleuchtungsverhältnisse anzupassen. So müssen Sie Ihre Augen nicht anstrengen, wenn Sie Ihren Computer in einer dunklen Umgebung betreiben.

Mithilfe von Umgebungslichtsensoren können Anwendungen auch die Inhalte auf gute Erkennbarkeit hin optimieren, sodass sich Ihr Computer in einem weiteren Bereich von Betriebsumgebungen als je zuvor benutzerfreundlich einsetzen lässt.

Die Plattform für Windows-Sensoren und -Ortung bietet ein Standardverfahren, um Sensoren und Ortungsgeräte in Windows zu integrieren, sowie eine Standard-API, um diese Geräte ansprechen zu können. Das Fehlen derartiger Funktionalität ist ein schon lange bestehendes Problem, da Windows-Entwickler bisher spezifische Sensorhardware und Sätze von APIs auswählen mussten, um damit in ihren Anwendungen zu arbeiten. Da keine Standards existierten, war der Wechsel zwischen verschiedenen Anbietern recht teuer. Mit Windows 7 erhalten Anwendungen eine einheitliche, standardisierte Schnittstelle, um die Sensoren auf Daten zu verwenden, und der Benutzer kann steuern, wie Daten von diesen Sensoren in Anwendungen zugänglich gemacht werden.

In diesem Buch haben wir der Plattform für Windows-Sensoren und -Ortung zwei Kapitel gewidmet. Kapitel 9 bringt eine Einführung zu diesem Thema. Dabei konzentriert es sich auf die Kernarchitektur der Plattform und erläutert die Rolle der verschiedenen Komponenten. Als Nächstes beschreibt das Kapitel die Art und Weise, in der Entwickler sensorische Eingaben in ihre Anwendungen integrieren können, indem Sensoren ermittelt werden und programmgesteuert auf die von den Sensoren bereitgestellten Daten zugegriffen wird. Am Ende von Kapitel 9 wissen Sie, wie Sie sowohl die nativen C++ (COM)-APIs als auch die verwalteten .NET-APIs programmieren.

Kapitel 10 erläutert im Detail den Ortungsteil der Plattform. Die Windows-Ortungs-APIs, die einen dedizierten Satz von APIs auf hoher Abstraktionsebene verkörpern, bauen auf der Windows 7-Sensor-Plattform auf. Die Ortungs-APIs existieren, um eine ganz einfache Frage zu beantworten: »Wo bin ich oder was ist meine aktuelle Position?« Die Antwort ist einfach, weil sie als Satz von Werten mit Länge, Breite und einem Fehlerradius dargestellt wird. Mit genauen Echtzeitortsinformationen lässt sich viel anfangen – so können Sie ortsabhängige Dienste anbieten, die eine Suche nach Ihrer aktuellen Position optimieren können, oder eine Kartenanwendung, die Ihre aktuelle Position erfasst und diese verfolgt, wenn Sie unterwegs sind und Ihren Standort wechseln. Kapitel 10 vermittelt Ihnen die erforderlichen Kenntnisse, um mithilfe der Ortungs-APIs Ortungsfunktionen in Ihre Anwendung zu integrieren.

Menüband

Wenn Sie das Programm Microsoft Paint in Windows 7 starten, was bemerken Sie zuerst? Was hat sich geändert seit Windows Vista (oder sollten wir sagen »seit Windows XP«)? Starten Sie nun WordPad. Was sehen Sie? In Windows 7 gibt es neue Versionen der bekannten Anwendungen Paint (siehe Abbildung 1.4) und WordPad (siehe Abbildung 1.5).

Beide Anwendungen bringen neue Features mit wie zum Beispiel die Fähigkeit, mit den Fingern zu malen (auf einem Multitouch-Gerät) oder ein *.docx*-Dokument von Word 2007 zu öffnen, ohne Microsoft Office Word 2007 auf dem Computer installiert zu haben. Doch vor allem springt die neue Benutzeroberfläche mit dem Menüband ins Auge.

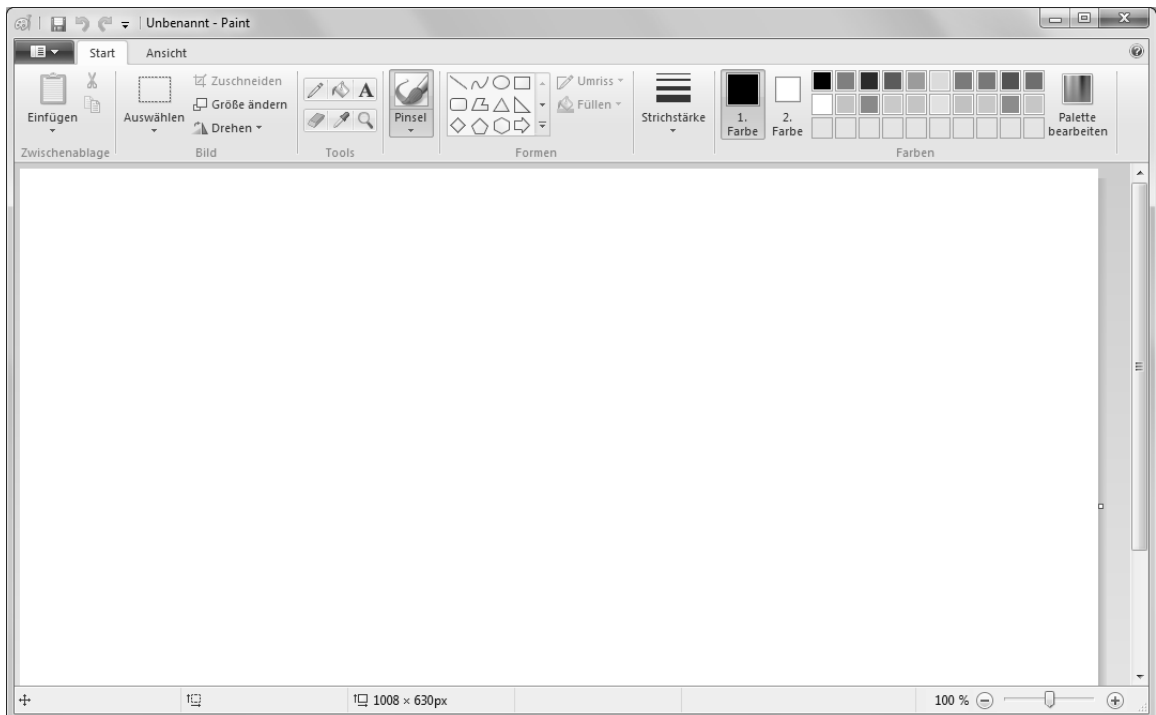


Abbildung 1.4 Das neue Erscheinungsbild von Paint

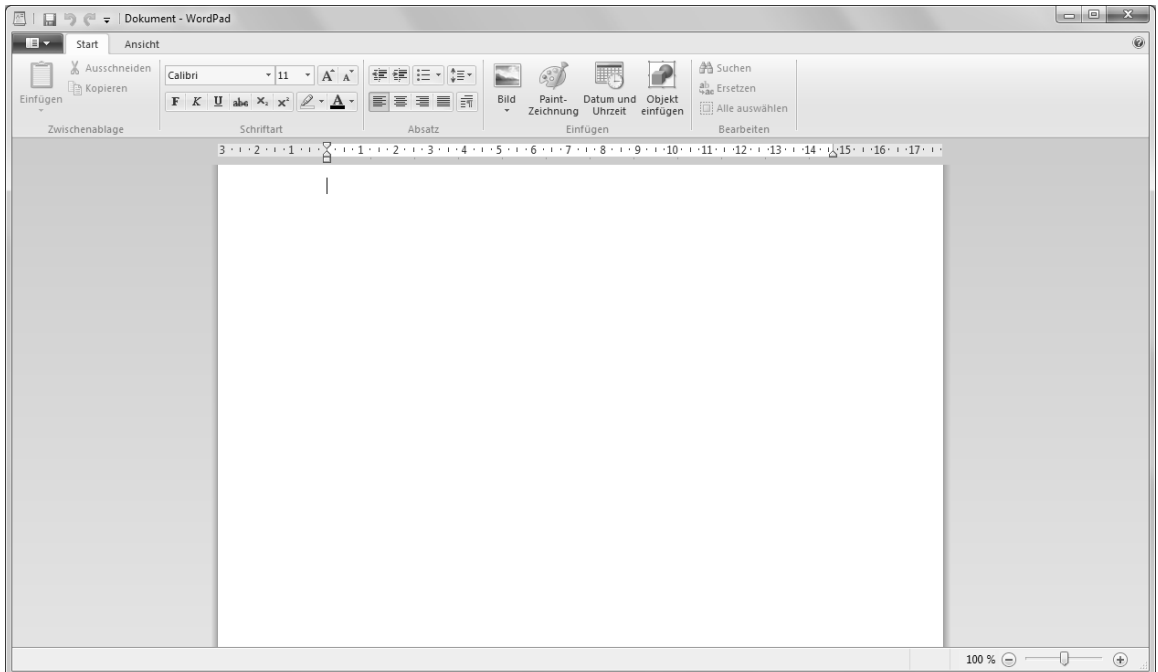


Abbildung 1.5 Das neue Erscheinungsbild von WordPad

Wir leben in einer Zeit, in der Software allgegenwärtig ist und von jedem verwendet wird. Der 3 Jahre alte Sohn von Alon sieht sich TV-Sendungen über Windows Media Center an. Seine Mutter schickt E-Mails an ihre Freunde und sein Heimautomatisierungssystem ist dafür programmiert, den Fernseher zu steuern und E-Mails zeitgerecht zu verschicken – und es weckt ihn auch jeden Morgen, indem es die Jalousien an seinen Schlafzimmerfenstern hochzieht. Die heutige Software ist wesentlich komplizierter als früher. Es gibt viel mehr Features, wir können mehr damit anstellen und wir können sie einsetzen, um unser Potenzial auszuschöpfen. Allerdings hat sich die von den meisten Windows-Anwendungen gebotene Benutzeroberfläche und vor allem die Benutzerfreundlichkeit nicht wesentlich geändert, um alle diese neuen Features und Anwendungsfälle von Software zu unterstützen. Es ist so, als ob man mit dem Lenkrad und den Pedalen eines Autos versucht, ein Flugzeug zu fliegen!

Die erste Gruppe innerhalb von Microsoft, die erkannt hat, dass die herkömmliche Benutzeroberfläche, die auf Menüs und Symbolleisten basiert, gescheitert ist, war das Office-Entwicklungsteam. Mit dem neuen Release von Office wurden neue Features hinzugefügt, doch hat niemand sie gefunden oder verwendet. Das Produkt ist zunehmend komplizierter geworden, die Benutzeroberfläche hat sich immer mehr aufgebläht. Die Benutzer fühlten, dass sie mehr tun konnten, doch wussten sie nicht, wie.

In Office 2007 nahm das Office-Team einen anderen, neueren Ansatz für den Entwurf der Benutzeroberfläche (User Interface, UI) – anstatt über Befehle nachzudenken, dachten sie über Features nach. Sie nennen es ergebnisorientiertes Design. Die Multifunktionsleiste (Ribbon) war geboren und mit ihr eine neue Benutzerfreundlichkeit. Mit der Leistung der modernen CPUs kombiniert mit visuellen Galerien, kann die Multifunktionsleiste das Ergebnis einer Aktion zeigen, wenn der Mauszeiger über den Befehlssymbolen schwebt. Möchten Sie ein Format festlegen, setzen Sie einfach den Mauszeiger über die einzelnen Formatsymbole in der Formatgalerie. Erhält ein Format den Fokus, erscheint eine Vorschau, wie das Dokument mit der Ände-

ung aussehen würde. Der Benutzer muss lediglich das Format auswählen, mit dem das Dokument am besten aussieht. Durch das featurebasierte Design lässt sich mit Office 2007 produktiver als mit jeder älteren Version von Office arbeiten.

HINWEIS Das im Englischen mit *Ribbon* bezeichnete *Menüband* von Windows 7 heißt in der deutschen Version von Office 2007 noch *Multifunktionsleiste*. Diese unterschiedliche Namensgebung ist lediglich eine Frage der Lokalisierung und wirkt sich weder auf das Erscheinungsbild noch die Funktionalität aus.

Wie immer ist das Office-Team der Vorreiter bei Benutzeroberflächen gewesen. Nachahmer finden Sie in den UI-Elementen anderer Microsoft-Produkte wieder, beispielsweise in Form der neuen Menüband-Unterstützung in den Microsoft Foundation Classes (MFC) von Visual Studio 2008 SP1 oder im Menüband-Toolkit für WPF und dem mit Windows 7 eingeführten Windows-Menübandframework. Es ist niemals so einfach gewesen, eine reichhaltige Benutzeroberfläche mit guter Benutzerfreundlichkeit in Win32-basierten Anwendungen bereitzustellen.

Viele moderne Frameworks für Benutzeroberflächen trennen den Benutzeroberflächenteil vom Code, der UI-Ereignisse behandelt und die Geschäftslogik steuert. Im Fall des Windows-Menübandframeworks ist die UI mithilfe von XML-basierter Markupsprache geschrieben, die in vielerlei Hinsicht der WPF XAML ähnelt, während die Codebehind-Kompilierungseinheit COM als Schnittstellentechnologie verwendet.

Kapitel 11 legt die Grundlagen für das Verständnis und die Verwendung des Windows-Menübandframeworks. Dieses Kapitel beschäftigt sich mit den Menübandbausteinen. Dabei lernen Sie, wie Sie das Menüband mithilfe seiner Markupsprache erstellen und wie Sie ihr Größenänderungsverhalten festlegen. Außerdem machen Sie sich mit den verschiedenen Menübandsteuerelementen vertraut. Als Nebeneffekt werden Sie ein besserer Benutzer des Menübands!

In Kapitel 12 sehen Sie, wie Sie mit dem Windows-Menübandframework programmieren, wie Sie das Menüband initialisieren und laden, wie Sie damit interagieren, wie Sie den Anwendungsstatus in der Benutzeroberfläche widerspiegeln und wie Sie das Konzept der Office Fluent-Benutzeroberfläche unterstützen.

Selbst wenn Sie ein .NET-Entwickler sind, der das Windows-Menübandframework nicht verwendet, empfehlen wir Ihnen, das einschlägige Kapitel zu lesen, um ein besseres Verständnis für dieses neue Konzept der Benutzeroberfläche zu bekommen. Dann greifen Sie basierend auf der Technologie, die Sie verwenden (Web, WPF oder Windows Forms), die am besten geeignete Menübandsteuerelementsuite heraus. Entwickler von nativem Code können ihre alten Anwendungen hernehmen und sie mit dem Glanz von Windows 7 versehen!

Erweiterte Grafikplattform

Dieses Buch enthält keine Kapitel zum neuen Grafikstack von Windows 7. Es sah so aus, als würden zu viele Informationen anfallen, die sich nicht in ein, zwei oder vielleicht drei Kapiteln unterbringen lassen, und wir wollten nicht einen erheblichen Teil dieses Buchs in einen Kurs der Art »Entwickeln von Windows 7-Grafik« umwandeln. Unabhängig davon bietet der neue DirectX-Grafikstack in Windows 7 Innovationen und großartige Tools für Entwickler, um leicht hardwarebeschleunigte 2D- und Textanwendungen zu erstellen, die auf den DirectX-APIs aufsetzen. Wir haben uns deshalb entschlossen, wenigstens in einem groben Überblick etwas über die Erweiterungen der Grafikplattform in Windows 7 zu schreiben.

Vor Windows 7

Das Microsoft Windows Graphics Device Interface (GDI) erlaubt Anwendungen, Grafiken und formatierten Text sowohl in der Videoanzeige als auch auf dem Drucker zu verwenden. Windows-basierte Anwendungen greifen nicht direkt auf die Grafikhardware zu. Stattdessen interagiert GDI mit Gerätetreibern im Namen der Anwendungen. Seit Windows 3.0 haben Entwickler mit GDI gearbeitet und seit dem Release von Windows XP haben unzählige Anwendungen GDI+ genutzt, einen vollständig neuen Satz von APIs. Selbst höhere Abstraktionsschichten wie zum Beispiel das .NET-WinForms-Rendering basieren auf GDI+. Windows-Anwendungsentwickler arbeiten schon geraume Zeit mit DirectX, um hochqualitative, hardwarebeschleunigte 3D-Grafiken bereitzustellen. Mit dem Debüt der Technologie im Jahre 1995 konnten Entwickler hochqualitative 3D-Grafiken für Spiele und technische Anwendungen für Spieler und Experten bereitstellen, die bereit waren, zusätzliches Geld für eine 3D-Grafikkarte auszugeben. Heute bringen selbst die billigsten PCs entsprechende 3D-Grafikhardware mit. Das WPF-Renderingmodul verwendet DirectX und erlaubt es Entwicklern von verwaltetem Code, hardwarebeschleunigte Grafikanwendungen nach dem neuesten Stand der Technik zu schreiben.

In Windows Vista wurde das Windows Display Driver Model (WDDM) eingeführt. Die WDDM-Infrastruktur für DirectX ermöglichte mehreren Anwendungen und Diensten, die Ressourcen der GPU (Graphical Processing Unit) gemeinsam zu nutzen. Der Desktop Window Manager (DWM) verwendet diese Technologie, um Taskwechsel in 3D zu animieren, um dynamische Miniaturansichten von Anwendungsfenstern bereitzustellen und die Windows Aero-Glaseffekte für Desktopanwendungen zu realisieren.

Neu in Windows 7

In Windows 7 hat Microsoft einen modernen Satz von APIs eingeführt, mit dem sich GDI+ hoffentlich endgültig ersetzen lässt. Windows 7 gibt mehr Grafikfähigkeiten in die Hände von Anwendungsentwicklern. Über einen neuen Satz von DirectX APIs, Win32 und verwaltetem Code können Entwickler von den letzten Neuerungen in GPUs profitieren, um ihren Anwendungen schnelle, skalierbare, hochqualitative 2D- und 3D-Grafiken, Text und Bilder hinzuzufügen. Alle diese neuen Technologien wurden dafür konzipiert, mit GDI und GDI+ zusammenzuarbeiten. Dabei wird zum einen gewährleistet, dass Entwickler ihre vorhandenen Investitionen problemlos bewahren können, und zum anderen ein klarer Migrationspfad geschaffen.

Diese erweiterten Grafikfähigkeiten werden durch die folgenden COM-basierten APIs realisiert (wobei einige auch in verwaltetem Code über das Windows API Code Pack verfügbar sind):

- Direct2D für das Zeichnen von 2D-Grafiken
- DirectWrite für das Anordnen und Rendern von Text
- Windows Imaging Component (WIC) für die Verarbeitung und Anzeige von Bildern
- Direct3D 10 für das Zeichnen von 3D-Grafiken
- Direct3D 11 für das Zeichnen von 3D-Grafiken mit Zugriff auf die GPU-Technologien der nächsten Generation wie zum Beispiel Mosaikschemas, begrenzte Unterstützung für Textur-Streaming und Allzweck-Computing
- DirectX Graphics Infrastructure (DXGI) für die Verwaltung von Geräten und GPU-Ressourcen und das Bereitstellen von Interoperabilität zwischen DirectX und GDI

Direct2D baut direkt auf Direct3D auf und bietet dem Win32-Entwickler auflösungsunabhängige 2D-APIs im Direktmodus, die die Leistung der Grafikhardware nutzen. Direct2D erlaubt hochqualitatives 2D-Rendering mit einer Leistung, die GDI und GDI+ überlegen ist, selbst wenn das Rendering ausschließlich durch Software und ohne Hardwarebeschleunigung erfolgt. Es gibt Entwicklern von Win32- und verwaltetem Code genauere Kontrolle über Ressourcen und ihre Verwaltung sowie eine höhere Abstraktion von den komplexen DirectX-APIs.

DirectWrite Ein großer Teil der Windows-Anwendungen richtet sich an Benutzer weltweit. Das bedeutet oftmals die Verwendung von mehreren Sprachen und verlangt nach einer Technologie, die hochqualitatives Textrendering, Auflösungsunabhängigkeit und Unicode-Text und -Layout unterstützt. Diese und andere Features werden nicht von GDI und GDI+ bereitgestellt, sondern von DirectWrite, einer neuen DirectX-Komponente mit folgenden Fähigkeiten:

- Hochqualitatives Teilpixel-, ClearType-Textrendering, das GDI, Direct2D oder anwendungsspezifische Ausgabetechnologien verwenden kann
- Hardwarebeschleunigter Text in Verbindung mit Direct2D
- Ein geräteunabhängiges Textlayoutsysteem, das die Texterkennbarkeit in Dokumenten und in der Benutzeroberfläche verbessert
- Unterstützung für Text in mehreren Formaten
- GDI-kompatibles Layout und Rendering
- Unterstützung für erweiterte typografische Features

Das DirectWrite-Schriftartsystem ermöglicht die Schriftnutzung nach dem Motto »jede Schrift, überall«, wobei Benutzer keine getrennten Installationsschritte ausführen müssen, nur um eine Schrift verwenden zu können. Es verfügt auch über eine verbesserte strukturelle Hierarchie für Schriftgruppierungen, um die manuelle oder programmgesteuerte Schriftermittlung zu unterstützen. Die APIs unterstützen Messen, Zeichnen und Treffertests von Mehrformattext, wie Abbildung 1.6 zeigt.



Abbildung 1.6 Ein Beispiel für Text, der mithilfe von DirectWrite gezeichnet wird

Die neue Grafik-API in Windows 7 bietet vielen vorhandenen GDI+-basierten Anwendungen einen klaren und entwicklerfreundlichen Pfad zu den neueren APIs.

Verbesserte Grundlagen

Die in den vorherigen Abschnitten beschriebenen Features könnten Sie an der Leistung von Windows 7 zweifeln lassen. Es stellt sich die Frage, wie sich so viele zusätzliche Funktionalität in das Betriebssystem integrieren lässt, ohne die Leistung auch nur im Mindesten zu beeinträchtigen. Zweifellos verlangt jedes zusätzliche Feature ein weiteres Aufrüsten des RAM im Computer, einen schnelleren Prozessor und eine größere Festplatte.

In der Tat besitzt Windows 7 dieselben Hardwareanforderungen wie Windows Vista und es läuft auf jeder Hardware schneller. Ich schreibe diese Zeilen, während bei mir ein RTM-Build von Windows 7 auf meinem 9-Zoll-Netbook läuft, das von einem einzigen, Low-End-900-MHz-Prozessor und 1 GB physischem RAM »getrieben« wird. Es ist mit einer äußerst langsamen 16 GB Solid-State-Festplatte ausgerüstet, die sogar das Internet-Browsen unmöglich machte, als ich Windows XP installiert hatte. Das betreffende Netbook führt nun Windows 7 seit dem Release Candidate aus und ich bin nie glücklicher mit seiner Leistung gewesen.

Dies zeugt von den großen Bemühungen der Windows-Entwickler, die Leistung und Zuverlässigkeit des Systems in Bezug auf nahezu jedes Feature des Betriebssystems zu verbessern. Windows 7 bootet schneller, geht schneller in den Standby-Modus, in den Ruhezustand und fährt schneller herunter als je zuvor. Windows 7 läuft mehrere Stunden mit 0% CPU-Nutzung und kaum bemerkbarem Speicherverbrauch, es benötigt weniger Leistung und spart mehr Systemressourcen als seine Vorgänger.

Doch halt! Da ist noch mehr! Windows 7 unterstützt 256 Kerne, was Ihnen einige Zeit verschafft, bevor Sie auf Hardware treffen, deren Kapazität das Betriebssystem nicht voll nutzen kann. Windows 7 profitiert von den neuesten Fortschritten in der Prozessor- und Speicherarchitektur – dem Non-Uniform Memory Access (NUMA)-Paradigma – und bringt Skalierbarkeitsverbesserungen für das gesamte Board. Außerdem weist Windows 7 zusätzliche Verbesserungen an der Leistungsverwaltung und dem Energieverbrauch auf, sodass Ihr mobiler PC einen ganzen Tag ohne Netzstrom auskommt. Darüber hinaus laufen zu jedem Zeitpunkt weniger Hintergrunddienste und einige Tasks werden nur gestartet, wenn sie wirklich benötigt werden – wenn Sie zum Beispiel die Verbindung zu einem Netzwerk herstellen oder wenn Sie Ihrem System ein externes Digitalisierungsgerät, eine Bluetooth-Komponente oder einen USB-Dongle hinzufügen.

In der Tat bestanden zwei der Hauptforderungen von Windows-Benutzern in den letzten Jahren darin, dass das System schneller läuft und weniger Ressourcen verbraucht. Dies lässt sich von Seiten des Betriebssystems allein nicht realisieren – es verlangt Kooperation und harte Arbeit von Ihnen, den Entwicklern, die Software für Windows 7 schreiben. Anders als die erstaunlichen Anstrengungen, die in den Windows-Kernel geflossen sind, um die Leistung zu verbessern, gibt es einige Dinge, die Sie wissen müssen, um sicherzustellen, dass Ihre Benutzer die gleiche Benutzerfreundlichkeit mit Ihren Anwendungen erfahren wie mit einer Out-of-the-Box-Installation von Windows 7.

Windows 7 lässt Sie bei dieser gewaltigen Aufgabe nicht allein. Es gibt großartige Tools, die wir im abschließenden Kapitel dieses Buchs erkunden und die Ihnen beim Profiling, der Fehlerbehebung und der Leistungsüberwachung Ihrer Anwendungen helfen. Windows 7 arbeitet großartig mit dem Windows Performance Toolkit, das als Profiler und Leistungsanalysator dient. Windows 7 baut auf der Grundlage auf, die durch Windows Vista gelegt wurde, mit Verbesserungen an Leistungsüberwachung und dem Zuverlässigkeitsmonitor. Windows 7 unterstützt Debugging-Techniken, die es erleichtern, Fehler in Ihrer Anwendung vor Ort zu beseitigen. Windows 7 führt sogar eine Problembehebungsplattform ein, um sicherzustellen, dass Ihre Benutzer in der Lage sind, in eigener Regie Probleme zu lösen – und wenn sie es nicht tun, ist immer die Problemaufzeichnung zur Stelle, damit die Benutzer das Problem reproduzieren und an Sie schicken können, um es zu analysieren. Die Abbildungen 1.7 bis 1.9 zeigen einige dieser Windows 7-Tools.

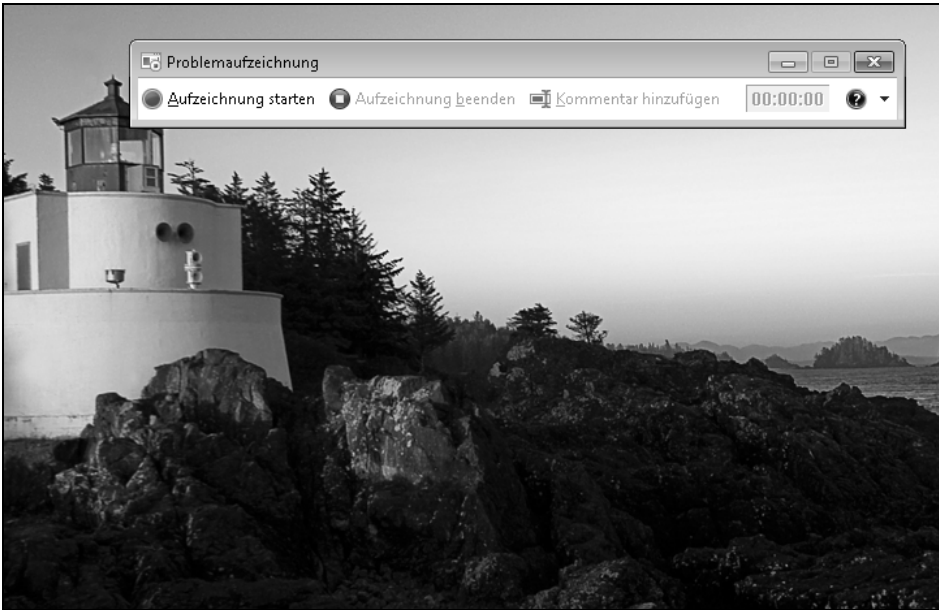


Abbildung 1.7 Die Problemaufzeichnung

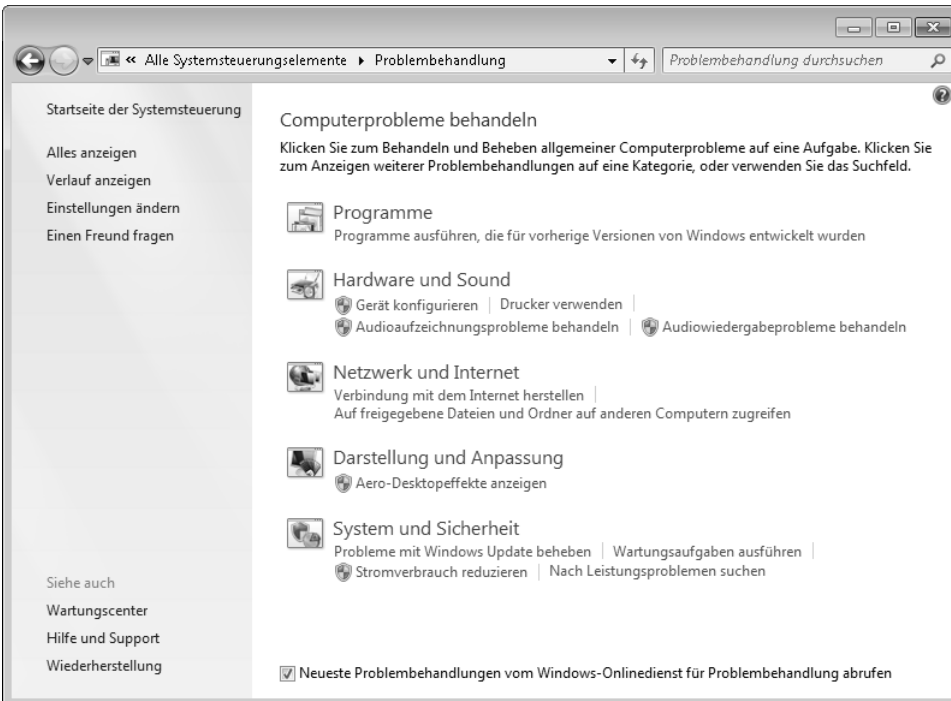


Abbildung 1.8 Problembehandlung von Computerproblemen

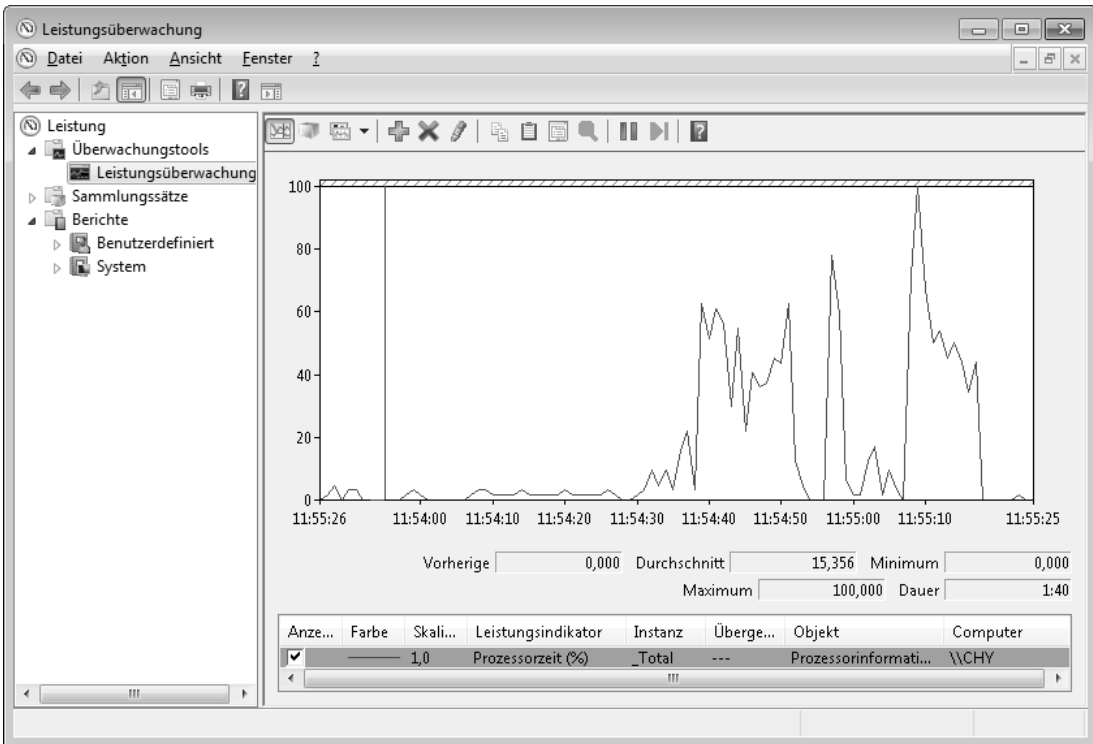


Abbildung 1.9 Windows-Leistungüberwachung

Und wenn Sie schließlich über eine ältere Anwendung stolpern, die nicht recht unter Windows 7 arbeitet, können Sie immer noch den virtuellen Windows XP-Computer verwenden, der in das Betriebssystem integriert ist. Eine Windows XP-Anwendung können Sie mit dem bekannten und funktionierenden Windows XP-Erscheinungsbild auf Ihrem Windows 7-Desktop ausführen lassen.

Zusammenfassung

Willkommen zu Windows 7! Das Buch, das Sie in Ihren Händen halten, ist das Ergebnis Hunderter Stunden Arbeit der Autoren, um die Geheimnisse der neuen APIs zu lüften und spezifische, greifbare und nachvollziehbare Lösungen zusammenzustellen, sodass Sie die Verwendung der neuen Features in Windows 7 schnell meistern können.

Wir hoffen, dass dieses Kapitel Ihnen Appetit darauf gemacht hat, mehr über die wichtigen Features von Windows 7 zu erfahren. Nach diesem Überblick geht es nun in medias res. Den Anfang macht die neue Windows 7-Taskleiste: Im nächsten Kapitel lernen Sie, wie Sie sie einsetzen können, um Ihre Windows 7-Anwendungen aufzupolieren!