



Windows-Standardprogramme per Code ermitteln

Wie es dem Nutzer beliebt

Die aktuellen Service Packs für Windows 2000 und XP versetzen den Nutzer in die Lage, für bestimmte Anwendungen wie Webbrowser oder E-Mail seine favorisierte Software anzugeben. Um diese in Ihren eigenen Anwendungen berücksichtigen zu können, bietet sich eine Klasse an, die diese Programme ermittelt und korrekt startet.

Der Kunde ist König. Und deshalb darf er seit Windows XP Service Pack 1 (SP1) und Windows 2000 Service Pack 3 (SP3) selbst entscheiden, welchen Internet-Browser, E-Mail-Client, Media-Player und so weiter er als Standardprogramme verwenden möchte. Die entsprechenden Einstellungen sind über *Start/Einstellungen/Systemsteuerung/Software* zu finden und stehen in diesem Dialog unter *Programmzugriff und -standards festlegen* zur Verfügung (siehe Abbildung 1). So weit, so gut. Dumm für den Anwender ist es, wenn sich Ihr Programm nicht daran hält.

Das Problem

Gesetzt den Fall, Sie möchten aus einem Programm heraus eine Internet-Seite im Browser öffnen. Nichts leichter als das. Fol-

Auf einen Blick

Autor

Neno Loje ist Microsoft Student Partner an der Universität Hamburg und Programmierer bei der Firma Keep it simple. Als unabhängiger Technologieberater schreibt er in Fachzeitschriften und hält Vorträge auf Konferenzen. Bei Codezone ist er Site-Editor für Deployment und Experte für Visual C# und die .NET Framework Base Class Library. Sie erreichen ihn über seine Homepage www.dotnet-online.de.

dotnetpro.code
A0506Zugriff



Technik Registry

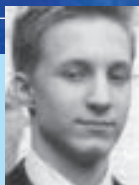


Tabelle 1

Für diese Client-Typen lassen sich in Windows Standardanwendungen festlegen.

Schlüssel	Beschreibung	Microsoft-Anwendung
Calendar	Kalender	Outlook
Contacts	Kontakte/Adressbuch	Outlook
IM	Instant Messenger	MSN Messenger
InternetCall	Internet-Telefonie	NetMeeting
JavaVM	Java Virtual Machine	Microsoft Virtual Machine
Mail	E-Mail-Client	Outlook (Express)
Media	Multimedia-Player	Windows Media Player
News	Newsreader	Outlook (Express)
StartMenuInternet	Internet-Browser	Internet Explorer

gender Einzeiler erledigt dies:

```
System.Diagnostics.Process.Start(
"http://www.dotnet-online.de");
```

Ist zu dem Zeitpunkt, zu dem diese Zeile ausgeführt wird, bereits ein Browser-Fenster geöffnet, so wird dieses zur Anzeige verwendet. In der Regel ist dies aber ein eher unglückliches Verhalten, da der Benutzer sicherlich einen Grund hatte, dieses Fenster nicht zu schließen.

Typische Fälle sind, dass er dort noch ein Formular bearbeiten will oder die angezeigten Informationen noch benötigt, etwa die eingegangenen E-Mails des Webpostfaches. Reißt Ihr Programm nun rücksichtslos das Fenster an sich, sind möglicherweise schon eingegebene Daten verloren.

Also muss eine andere Methode her. Sofern der Entwickler sicher sein kann, dass der Internet Explorer auf dem Rechner zur Verfügung steht, lässt sich das Öffnen in einem neuen Fenster erzwingen, indem er explizit einen neuen Internet-Explorer-Prozess von *iexplore.exe* erstellt.

```
System.Diagnostics.Process.Start(
"iexplore.exe", "http://www.dotnet-online.de");
```

Diese Anweisung führt zum gewünschten Effekt. Trotzdem sind Magenschmerzen beim Benutzer vorprogrammiert – nämlich falls dieser sich zu einem anderen Browser hingezogen fühlt als dem Internet Explorer und diesen korrekt im Dialog *Programmzugriff und -standards festlegen* eingestellt hat. Dummerweise bieten weder das Windows-API noch das .NET Framework entsprechende Klassen, um die vom Nutzer getroffenen Einstellungen zu ermitteln. Hier ist also Handarbeit angesagt. Trotzdem bleibt es bei einem Einzeiler, der zum Beispiel so aussieht:

```
DefaultClients.Start(ClientType.InternetBrowser,
"http://www.dotnet-online.de");
```

Voraussetzung für diesen kurzen und knackigen Befehl ist die Klasse *DefaultClients*, die Sie nun kennen lernen.

Die Registry als Informationsquelle

Ausgangspunkt ist die Registry, die alle wichtigen Informationen betrifft Windows verwaltet. Dazu gehört auch eine Liste mit Programmen zu verschiedenen Client-Typen wie Webbrowser, E-Mail-Client und so

Den vollständigen Artikel lesen Sie in:



dotnetpro 06/2005 auf Seite 104

dotnetpro-Abonnenten können diesen über das Online-Archiv herunterladen:

<http://www.dotnetpro.de/articles/onlinearticle1695.aspx>