Erstellen sicherer ASP.NET-Anwendungen

Authentifizierung, Autorisierung und sichere Kommunikation

Auf der Orientierungsseite finden Sie einen Ausgangspunkt und eine vollständige Übersicht zum *Erstellen sicherer ASP.NET-Anwendungen*.

Zusammenfassung

Webdienste unterstützen die Clientzertifikatsauthentifizierung als ein Mittel zur Authentifizierung von Clientanwendungen. Nachstehend wird erläutert, wie ein Webdienst für die Clientzertifikatsauthentifizierung konfiguriert wird. Darüber hinaus erfahren Sie, wie der Webdienst aufgerufen und ein Zertifikat von einer ASP.NET-Webanwendung übergeben wird.

Vorgehensweise: Aufrufen eines Webdienstes mit Clientzertifikaten von ASP.NET

Webdienste müssen häufig über die Möglichkeit verfügen, ihre Aufrufer (andere Anwendungen) zu authentifizieren, um eine Autorisierung durchzuführen. Mit Clientzertifikaten steht ein ausgezeichneter Authentifizierungsmechanismus für Webdienste zur Verfügung. Wenn Sie mit Clientzertifikaten arbeiten, profitiert die Anwendung zudem von der Errichtung eines sicheren Kanals (unter Verwendung von Secure Sockets Layer (SSL)) zwischen Clientanwendung und Webdienst. Hiermit sind Sie in der Lage, auf sichere Weise vertrauliche Informationen an den Webdienst zu übermitteln und von diesem abzurufen. SSL gewährleistet die Vertraulichkeit und die Integrität von Nachrichten.

In dieser Vorgehensweise wird erläutert, wie ein Webdienst aufgerufen wird, der für die Anforderung von Clientzertifikaten konfiguriert ist.

Hinweis: Die Informationen in dieser Vorgehensweise gelten auch für Remotekomponenten, für die ASP.NET und IIS als Host fungieren.

Gründe für die Verwendung einer Serviced Component

Bei der hier vorgestellten Lösung wird eine Serviced Component verwendet, die für die Ausführung in einer Enterprise Services-Serveranwendung unter Verwendung eines benutzerdefinierten Dienstkontos konfiguriert ist. Die ASP.NET-Webanwendung ruft die Serviced Component auf, die ihrerseits wiederum den Webdienst aufruft (und ein Clientzertifikat übergibt). Die Konfiguration der Lösung ist in Abbildung 1 dargestellt.

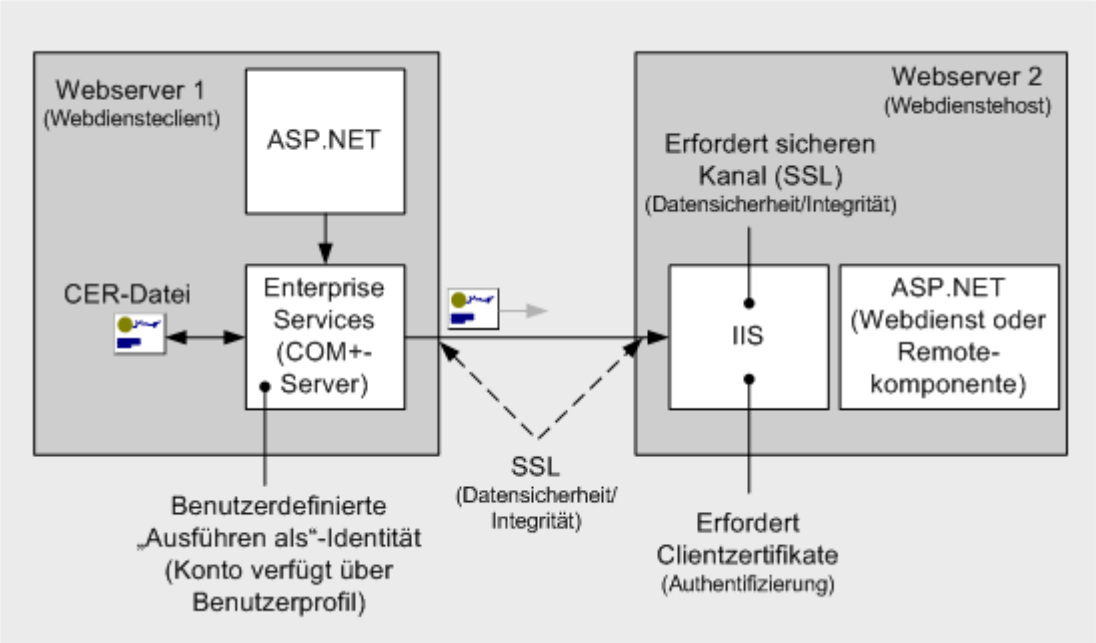

Abbildung 1
ASP.NET ruft eine Serviced Component auf, die wiederum den Webdienst aufruft

Mit diesem Ansatz wird sichergestellt, dass das System bei der Kommunikation mit dem Webdienst auf ein Benutzerprofil zugreifen kann. Dies ist für den anfänglichen SSL-Handshake zwingend erforderlich.

Hinweis: Bei dem ASPNET-Konto, das für das Ausführen von Webanwendungen herangezogen wird, ist die Berechtigung zum "Verweigern der interaktiven Anmeldung" aktiviert, wodurch eine interaktive Anmeldung mit diesem Konto verhindert wird. Infolgedessen verfügt das ASPNET-Konto nicht über ein Benutzerprofil.

Sie sollten dem ASPNET-Konto (oder einem beliebigen anderen Konto, das zum Ausführen von Webanwendungen verwendet wird) nicht die Fähigkeit zur interaktiven Anmeldung verleihen. Halten Sie sich bei der Konfiguration von Konten zum Ausführen von Webanwendungen immer an das Prinzip der minimalen Rechte, und gewähren Sie nur die Rechte, die unbedingt erforderlich sind. Weitere Informationen hierüber finden Sie unter "Vorgehensweise: Erstellen eines benutzerdefinierten Kontos zum Ausführen von ASP.NET" im Abschnitt "Referenz" dieses Handbuchs.

Gründe für die Verwendung eines Benutzerprofils

Wenn Sie eine Anforderung an einen Webdienst senden, der ein Clientzertifikat voraussetzt, erfolgt ein SSL-Handshake zwischen Client und Server. Zu den Elementen, die hierbei ausgetauscht werden, gehört ein Serverzertifikat, ein Clientzertifikat und ein "vorläufiger geheimer Masterschlüssel", der clientseitig erzeugt wird. Dieser geheime Schlüssel wird an späterer Stelle im Protokoll verwendet, um einen "geheimen Masterschlüssel" zu erzeugen.

Damit der Server überprüfen kann, dass der Übermittler des Zertifikats auch tatsächlich der Inhaber des privaten Schlüssels ist, muss der Client den vorläufigen geheimen Masterschlüssel mit dem privaten Schlüssel verschlüsseln und den verschlüsselten vorläufigen Masterschlüssel an den Server zurücksenden. Damit das System nun auf den privaten Schlüssel des Clients zum Signieren des vorläufigen Masterschlüssels zugreifen kann, muss das System auf den Schlüsselspeicher des Clients zugreifen. Dieser Schlüsselspeicher befindet sich im Profil des Clients, das daher geladen sein muss.

Anforderungen

Im Folgenden finden Sie eine Liste der empfohlenen Hardware und Software und eine Beschreibung der Netzwerkinfrastruktur, der Fähigkeiten und Kenntnisse sowie der Service Packs, die Sie benötigen.

- Microsoft® Windows® 2000 als Betriebssystem
- Microsoft Visual Studio® .NET als Entwicklungssystem
- Zugriff auf eine Zertifizierungsstelle (Certificate Authority, CA) zwecks Erzeugung neuer Zertifikate
- Einen Webserver mit einem installierten Serverzertifikat
  Weitere Informationen über das Installieren von Webserverzertifikaten finden Sie unter "Vorgehensweise: Einrichten von SSL auf einem Webserver".

Die in dieser Vorgehensweise erläuterten Verfahren setzen zudem Kenntnisse des Entwicklungstools Microsoft Visual C#™ voraus.

Zusammenfassung

Diese Vorgehensweise enthält folgende Verfahren:

1. Erstellen eines einfachen Webdienstes
2. Konfigurieren des virtuellen Verzeichnisses des Webdienstes für die Anforderung von Clientzertifikaten
3. Erstellen eines benutzerdefinierten Kontos für die Ausführung einer Serviced Component
4. Anfordern eines Clientzertifikats für das benutzerdefinierte Konto
5. Testen des Clientzertifikats mit einem Browser
6. Exportieren des Clientzertifikats in eine Datei
7. Entwickeln der Serviced Component für den Aufruf des Webdienstes
8. Konfigurieren und Installieren der Serviced Component
9. Entwickeln einer Webanwendung zum Aufrufen der Serviced Component

---

Hinweis: In dieser Vorgehensweise trägt der Webdienstcomputer (der als Host für den Webdienst fungiert) den Namen "WSServer", und der Webdienst-Clientcomputer (der als Host für die Client-ASP.NET-Webanwendung und die Serviced Component fungiert) wird als "WSClient" bezeichnet.

---

1. Erstellen eines einfachen Webdienstes

▶ **So erstellen Sie einen einfachen Webdienst auf dem Webdienst-Hostcomputer**

1. Starten Sie Visual Studio .NET, und erstellen Sie mit C# eine neue ASP.NET-Webdienstanwendung mit Namen **SecureMath**.
2. Benennen Sie den Dienst **Service1.asmx** in **Math.asmx** um.
3. Öffnen Sie **Math.asmx.cs**, und benennen Sie die Klasse **Service1** in **math** um.
4. Fügen Sie der **math**-Klasse die folgende **Web**-Methode hinzu.

```
[WebMethod]
public long Add(long operand1, long operand2)
{
    return (operand1 + operand2);
}
```

5. Klicken Sie im Menü **Erstellen** auf **Projektmappe erstellen**, um den Webdienst zu erstellen.

2. Konfigurieren des virtuellen Verzeichnisses des Webdienstes für die Anforderung von Clientzertifikaten

In diesem Verfahren wird IIS (Internet-Informationsdienste) verwendet, um das virtuelle Verzeichnis des Webdienstes für SSL und die Anforderung von Zertifikaten zu konfigurieren.

Das Verfahren setzt zudem voraus, dass auf dem Webserver ein gültiges Zertifikat installiert ist. Weitere Informationen über das Installieren von Webserverzertifikaten finden Sie im Abschnitt "Referenz" dieses Handbuchs unter "Vorgehensweise: Einrichten von SSL auf einem Webserver".

▶ **So konfigurieren Sie das virtuelle Verzeichnis des Webdienstes für die Anforderung von Clientzertifikaten**

1. Starten Sie die Internet-Informationsdienste auf dem Hostcomputer des Webdienstes.
2. Wechseln Sie zum virtuellen Verzeichnis **SecureMath**.
3. Klicken Sie mit der rechten Maustaste auf **SecureMath**, und klicken Sie dann auf **Eigenschaften**.
4. Klicken Sie auf die Registerkarte **Verzeichnissicherheit**.
5. Klicken Sie unter **Sichere Kommunikation** auf **Bearbeiten**.
   Wenn der Befehl **Bearbeiten** nicht zur Verfügung steht, ist wahrscheinlich kein Webserverzertifikat installiert.
6. Aktivieren Sie das Kontrollkästchen **Sicheren Kanal verlangen (SSL)**.
7. Wählen Sie die Option **Clientzertifikate verlangen**.
8. Klicken Sie auf **OK** und dann noch einmal auf **OK**.
9. Klicken Sie im Dialogfeld **Vererbungsüberschreibungen** auf **Alles auswählen**, und klicken Sie dann auf **OK**, um das Eigenschaftendialogfeld von **SecureMath** zu schließen.
   Hiermit werden die neuen Sicherheitseinstellungen allen Unterverzeichnissen im virtuellen Stammverzeichnis zugewiesen.

3. Erstellen eines benutzerdefinierten Kontos für die Ausführung der Serviced Component

Mit diesem Verfahren wird ein neues Benutzerkonto auf dem Webdienst-Clientcomputer erstellt, das Sie zum Ausführen der Serviced Component benötigen, die den Webdienst aufruft.

▶ **So erstellen Sie ein benutzerdefiniertes Konto für die Ausführung der Serviced Component**

1. Erstellen Sie auf dem Clientcomputer ein neues Benutzerkonto mit einem starken Kennwort. Deaktivieren Sie das Kontrollkästchen **Benutzer muss Kennwort bei der nächsten Anmeldung ändern**, und aktivieren Sie die Option **Kennwort läuft nie ab**.
2. Fügen Sie das Konto der Gruppe der Administratoren hinzu.
   Das Konto, das zum Laden eines Benutzerprofils verwendet wird, muss auf dem lokalen Computer der Gruppe der Administratoren angehören.

4. Anfordern eines Clientzertifikats für das benutzerdefinierte Konto

In diesem Verfahren melden Sie sich unter Verwendung des neuen benutzerdefinierten Kontos am Clientcomputer an. Anschließend geben Sie eine Zertifikatsanforderung aus. Bei diesem Verfahren wird davon ausgegangen, dass Sie die Microsoft Zertifikatdienste verwenden. Wenn Sie für die Erstellung neuer Zertifikate nicht auf die Microsoft Zertifikatdienste zurückgreifen, fordern Sie ein Clientzertifikat von Ihrer bevorzugten

Zertifizierungsstelle an, und installieren Sie das Zertifikat, während Sie mit dem benutzerdefinierten Konto angemeldet sind.

Bei diesem Verfahren wird außerdem davon ausgegangen, dass die Microsoft Zertifikatdienste so konfiguriert sind, dass in Reaktion auf Zertifikatsanforderungen automatisch Zertifikate ausgegeben werden. Die Microsoft Zertifikatdienste können auch für ausstehende Anforderungen konfiguriert sein, was bedeutet, dass das Zertifikat explizit von einem Administrator ausgegeben werden muss.

▶ **So prüfen Sie die Einstellung der Microsoft Zertifikatdienste**

1. Klicken Sie auf dem Computer, auf dem die Microsoft Zertifikatdienste ausgeführt werden, in der Programmgruppe Verwaltung auf **Zertifizierungsstelle**.
2. Erweitern Sie **Zertifizierungsstelle (Lokal)**, klicken Sie mit der rechten Maustaste auf die Zertifizierungsstelle, und klicken Sie dann auf **Eigenschaften**.
3. Klicken Sie auf die Registerkarte **Richtlinienmodul**, und klicken Sie dann auf **Konfigurieren**.
4. Aktivieren Sie die Standardaktion.

Im nachstehenden Verfahren wird vorausgesetzt, dass **Zertifikat immer ausstellen** ausgewählt ist.

▶ **So fordern Sie ein Clientzertifikat für das benutzerdefinierte Konto an**

1. Melden Sie sich vom Clientcomputer ab, und melden Sie sich unter dem benutzerdefinierten Konto wieder an.

   Hiermit wird die Erstellung eines Benutzerprofils für das benutzerdefinierte Konto erzwungen.
2. Wechseln Sie zur Zertifizierungsstelle, um ein Clientzertifikat anzufordern. Wenn sich die Zertifizierungsstelle beispielsweise auf dem CAServer-Computer befindet, geben Sie folgenden URL ein:

   ```
   http://caserver/certsrv
   ```

3. Klicken Sie auf **Zertifikat anfordern** und dann auf **Weiter**.
4. Vergewissern Sie sich, dass **Benutzerzertifikat** aktiviert ist, und klicken Sie dann auf **Weiter**.
5. Klicken Sie auf **Senden**.

   Nun wird eine Anforderung erzeugt und zwecks Verarbeitung an die Zertifizierungsstelle gesendet.
6. Nachdem das Zertifikat ausgestellt wurde und Sie eine Antwort vom CA-Server erhalten haben, klicken Sie auf **Dieses Zertifikat installieren**.
7. Stellen Sie sicher, dass auch das Zertifikat der ausgebenden Zertifizierungsstelle als vertraute Stammzertifizierungsstelle auf dem lokalen Computer installiert ist.

   Um sich zu vergewissern, führen Sie die folgenden Schritte durch:

   a. Klicken Sie auf der Taskleiste auf **Start** und dann auf **Ausführen**.

   b. Geben Sie **mmc** ein, und klicken Sie dann auf **OK**.

   c. Klicken Sie im Menü **Datei** auf **Snap-In hinzufügen/entfernen**.

   d. Klicken Sie auf **Hinzufügen**.

   e. Klicken Sie auf **Zertifikate** und dann auf **Hinzufügen**.

   f. Klicken Sie auf **Computerkonto** und dann auf **Weiter**.

   g. Klicken Sie auf **Lokaler Computer (Computer, auf dem diese Konsole ausgeführt wird)** und dann auf **Fertig stellen**.

   h. Klicken Sie auf **Schließen** und dann auf **OK**.

   i. Erweitern Sie im linken Fensterbereich des MMC-Snap-Ins den Eintrag **Zertifikate (Lokaler Computer)**.

j. Erweitern Sie **Vertrauenswürdige Stammzertifizierungsstellen**, und klicken Sie dann auf **Zertifikate**.

k. Vergewissern Sie sich, dass das Zertifikat Ihrer Zertifizierungsstelle aufgeführt wird.

Befindet sich das Zertifizierungsstellenzertifikat nicht in der Liste, führen Sie die folgenden Schritte durch:

a. Besuchen Sie **http://caserver/certsrv**.

b. Klicken Sie auf **Zertifizierungsstellenzertifikat oder Zertifikatsperrliste abrufen**, und klicken Sie dann auf **Weiter**.

c. Klicken Sie auf **Diesen Zertifizierungsstellen-Zertifikatspfad installieren.**

5. Testen des Clientzertifikats mit einem Browser

In diesem Verfahren zeigen Sie den Webservice in einem Browser an, um sich zu vergewissern, dass es keine Probleme mit den Server- oder Clientzertifikaten gibt.

▶ **So testen Sie das Clientzertifikat mit einem Browser**

1. Öffnen Sie Internet Explorer, und navigieren Sie zu **https://server/SecureMath/Math.asmx**.
   Vergewissern Sie sich, dass Sie "https" angegeben haben, da diese Site für die Verwendung von SSL konfiguriert ist.
2. Nun sollte das Dialogfeld **Clientauthentifizierung** angezeigt werden. Wählen Sie Ihr Clientzertifikat aus, und klicken Sie dann auf **OK**.
3. Vergewissern Sie sich, dass die Webdienst-Testseite im Browser ordnungsgemäß angezeigt wird.
   Wird das in Abbildung 1 dargestellte Dialogfeld angezeigt, müssen Sie das Zertifizierungsstellenzertifikat im Speicher vertrauenswürdiger Stammzertifizierungsstellen installieren, wie im vorstehenden Verfahren erläutert.

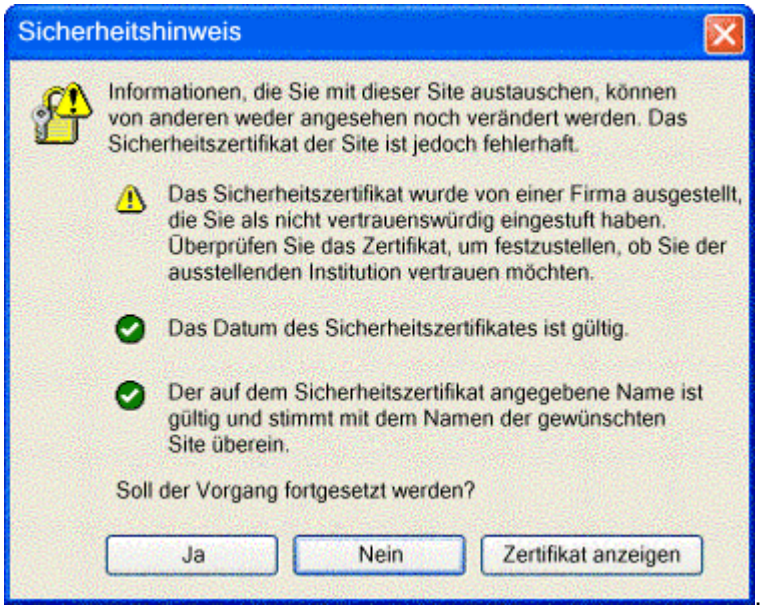

Abbildung 1
Dialogfeld "Security Alert" (Sicherheitshinweis)

6. Exportieren des Clientzertifikats in eine Datei

In diesem Verfahren wird das Clientzertifikat in eine Datei exportiert. Diese Datei wird anschließend von der Serviced Component abgerufen, wenn diese das Zertifikat an den Webdienst übergeben muss.

▶ **So exportieren Sie das Clientzertifikat in eine Datei**

1. Klicken Sie im Internet Explorer im Menü **Extras** auf **Internetoptionen**.
2. Klicken Sie auf die Registerkarte **Inhalt**.
3. Klicken Sie auf **Zertifikate**.
4. Klicken Sie auf das Clientzertifikat und dann auf **Exportieren**.
5. Klicken Sie auf **Weiter**, um den Begrüßungsdialog des Zertifikatexport-Assistenten zu schließen.
6. Vergewissern Sie sich, dass **Nein, privaten Schlüssel nicht exportieren** aktiviert ist, und klicken Sie dann auf **Weiter**.
7. Stellen Sie sicher, dass **DER-codiert-binär X.509 (.CER)** aktiviert ist, und klicken Sie dann auf **Weiter**.

   Sie müssen dieses Format verwenden, weil das .NET Framework Base-64- oder PKCS #7-Formate nicht unterstützt.
8. Geben Sie einen Namen für die Exportdatei ein. Notieren Sie die Position der Exportdatei mit der Endung CER, da Sie diese in einem nachfolgenden Verfahren noch benötigen.
9. Klicken Sie auf **Weiter**, und klicken Sie dann auf **Fertig stellen**, um das Zertifikat zu exportieren.
10. Schließen Sie Internet Explorer.
11. Melden Sie sich vom Computer ab, und melden Sie sich unter Ihrem regulären Entwicklerkonto wieder an.

7. Entwickeln der Serviced Component für den Aufruf des Webdienstes

In diesem Verfahren wird eine neue C#-Klassenbibliotheksanwendung sowie die Serviced Component erstellt, die für den Aufruf des Webdienstes benötigt wird. Das Verfahren setzt voraus, dass Sie auf dem Clientcomputer arbeiten.

▶ **So entwickeln Sie die Serviced Component für den Aufruf des Webdienstes**

1. Starten Sie Visual Studio .NET, und erstellen Sie in C# ein neues Klassenbibliotheksprojekt mit Namen **WebServiceRequestor**.
2. Fügen Sie dem Webdienst **SecureMath** einen Webverweis hinzu.

   Wichtig: Bevor Sie den Webverweis hinzufügen, müssen Sie das virtuelle Verzeichnis des Webdienstes vorübergehend so konfigurieren, dass keine Clientzertifikate verlangt werden (obwohl immer noch SSL vorausgesetzt wird). Nachdem Sie den Webverweis erfolgreich hinzugefügt haben, ändern Sie die Konfiguration des virtuellen Verzeichnisses wieder so, dass Clientzertifikate angefordert werden.

   Wenn eine Site Clientzertifikate erfordert, stellt der Herausgeber des Webdienstes in der Praxis die WDSL als separate Offlinedatei zur Verfügung, die die Nutzer (des Dienstes) dann verwenden können, um den Proxy zu erstellen.

   Stellen Sie sicher, dass im Dialogfeld **Webverweis hinzufügen** bei der Festlegung der Adresse des Webdienstes **https** angegeben wird. Andernfalls wird ein Fehler erzeugt, da das virtuelle Verzeichnis des Webdienstes für die Verwendung von SSL konfiguriert ist.
3. Fügen Sie einen Verweis auf die **System.EnterpriseServices**-Assembly hinzu.
4. Benennen Sie **Class1.cs** in **ProfileManager.cs** um.
5. Fügen Sie **ProfileManager.cs** die folgende Klassendefinition hinzu (wodurch das Klassengerüst **Class1** ersetzt wird). Die **ProfileManager**-Klasse verwendet **P/Invoke** zum Aufrufen der Win32-APIs **LoadUserProfile** und **UnloadUserProfile**.

```
internal class ProfileManager
{
  [DllImport("Userenv.dll", SetLastError=true,
            CharSet=System.Runtime.InteropServices.CharSet.Auto)]
  internal static extern bool LoadUserProfile(IntPtr hToken,
                                              ref PROFILEINFO lpProfileInfo);

  [DllImport("Userenv.dll", SetLastError=true,
            CharSet=System.Runtime.InteropServices.CharSet.Auto)]
  internal static extern bool  UnloadUserProfile(IntPtr hToken,
                                                 IntPtr hProfile);

  [StructLayout(LayoutKind.Sequential, CharSet=CharSet.Ansi)]
  public struct PROFILEINFO
  {
    public int dwSize;
    public int dwFlags;
    public String lpUserName;
    public String lpProfilePath;
    public String lpDefaultPath;
    public String lpServerName;
    public String lpPolicyPath;
    public IntPtr hProfile;
  }
}
```

6. Fügen Sie dem Projekt eine zweite Klassendatei mit Namen **MathServiceComponent.cs** hinzu.
7. Fügen Sie **MathServiceComponent.cs** unterhalb der vorhandenen **using**-Anweisung die folgenden **using**-Anweisungen hinzu.

```
using System.Net;
using System.Web.Services;
using System.Security.Principal;
using System.EnterpriseServices;
using System.Runtime.InteropServices;
using System.Security.Cryptography.X509Certificates;
using WebServiceRequestor.WebReference1;
```

8. Fügen Sie die folgende Klassendefinition hinzu, mit der eine **CallMathWebService**-Methode vom Typ **public** bereitgestellt wird. Sie rufen diese Methode in einem nachfolgenden Verfahren von einer Client-ASP.NET-Webanwendung aus auf.

```
// This class calls the web service that requires a certificate.
public class MathServiceComponent : ServicedComponent
{
  [DllImport("advapi32.dll", CharSet=CharSet.Auto, SetLastError=true)]
  private extern static bool DuplicateToken(IntPtr ExistingTokenHandle,
                                            int SECURITY_IMPERSONATION_LEVEL,
                                            ref IntPtr DuplicateTokenHandle);

  [DllImport("kernel32.dll", CharSet=CharSet.Auto)]
  private extern static bool CloseHandle(IntPtr handle);
```

```
// Calls the Web service that requires client certificates
// certFilepath points to the .cer file to use
// url is the Web service url
// operand1 and operand2 are the parameters to pass to the Web service
public long CallMathWebService(String certFilepath,
                               String url, int operand1, int operand2)
{
  bool retVal = false;
  // Need to duplicate the token. LoadUserProfile needs a token with
  // TOKEN_IMPERSONATE and TOKEN_DUPLICATE.
  const int SecurityImpersonation = 2;
  IntPtr dupeTokenHandle = DupeToken(WindowsIdentity.GetCurrent().Token,
                                     SecurityImpersonation);
  if(IntPtr.Zero == dupeTokenHandle)
  {
    throw new Exception("Unable to duplicate token.");
  }
  // Load the profile.
  ProfileManager.PROFILEINFO profile = new ProfileManager.PROFILEINFO();
  profile.dwSize = 32;
  profile.lpUserName = @"alexmlaptop\CustomASPNET";
  retVal = ProfileManager.LoadUserProfile(dupeTokenHandle, ref profile);
  if(false == retVal)
  {
    throw new Exception("Error loading user profile. " +
                        Marshal.GetLastWin32Error());
  }
  // Instantiate the Web service proxy
  math mathservice = new math();
  mathservice.Url = url;
  String certPath = certFilepath;
  mathservice.ClientCertificates.Add(
                              X509Certificate.CreateFromCertFile(certPath));
  long lngResult = 0;
  try
  {
    lngResult = mathservice.Add(operand1, operand2);
  }
  catch(Exception ex)
  {
    if(ex is WebException)
    {
      WebException we = ex as WebException;
      WebResponse webResponse = we.Response;
      throw new Exception("Exception calling method. " + ex.Message);
    }
  }
  ProfileManager.UnloadUserProfile(WindowsIdentity.GetCurrent().Token,
                                   profile.hProfile);
  CloseHandle(dupeTokenHandle);
  return lngResult;
}
```

```
  private IntPtr DupeToken(IntPtr token, int Level)
  {
    IntPtr dupeTokenHandle = new IntPtr(0);
    bool retVal = DuplicateToken(token, Level, ref dupeTokenHandle);
    if (false == retVal)
    {
      return IntPtr.Zero;
    }
    return dupeTokenHandle;
  }
} // end class
```

9. Klicken Sie im Menü **Erstellen** auf **Projektmappe erstellen**.

8. Konfigurieren und Installieren der Serviced Component

In diesem Verfahren wird die Serviced Component konfiguriert, es wird ein starker Name erzeugt, und die Serviced Component wird im globalen Assemblycache installiert und in COM+ registriert.

1. Öffnen Sie **Assemblyinfo.cs**, und fügen Sie unterhalb der vorhandenen **using**-Anweisungen die folgende **using**-Anweisung hinzu.

```
using System.EnterpriseServices;
```

2. Fügen Sie **Assemblyinfo.cs** das folgende Assemblyebenenattribut hinzu, um die Serviced Component für die Ausführung in einer COM+-Serveranwendung zu konfigurieren.

```
[assembly: ApplicationActivation(ActivationOption.Server)]
```

3. Öffnen Sie ein Befehlsfenster, und wechseln Sie in das aktuelle Projektverzeichnis.
4. Erzeugen Sie mithilfe des Dienstprogramms **Sn.exe** eine Schlüsseldatei, die ein Schlüsselpaar bestehend aus einem öffentlichen und einem privaten Schlüssel enthält.

```
sn.exe -k WebServiceRequestor.snk
```

5. Kehren Sie zu Visual Studio .NET zurück.
6. Suchen Sie in **Assemblyinfo.cs** das Attribut [**AssemblyKeyFile**], und ändern Sie es so, dass es auf die Schlüsseldatei im Projektverzeichnis verweist:

```
[assembly: AssemblyKeyFile(@"..\..\WebServiceRequestor.snk")]
```

7. Klicken Sie im Menü **Erstellen** auf **Projektmappe erstellen**.
8. Kehren Sie zur Befehlseingabeaufforderung zurück, und führen Sie den folgenden Befehl aus, um die Assembly dem globalen Assemblycache hinzuzufügen.

```
gacutil.exe /i bin\debug\webservicerequestor.dll
```

9. Führen Sie den folgenden Befehl aus, um die Assembly in COM+ zu registrieren.

```
regsvcs bin\debug\webservicerequestor.dll
```

10. Starten Sie die Komponentendienste (aus der Programmgruppe Verwaltung).

11. Erweitern Sie die Knoten **Komponentendienste**, **Computer** und **Arbeitsplatz**.
12. Erweitern Sie den Ordner **COM+-Anwendungen**.
13. Klicken Sie mit der rechten Maustaste auf **WebServiceRequestor**, und klicken Sie dann auf **Eigenschaften**.
14. Klicken Sie auf die Registerkarte **Identität**.
15. Wählen Sie die Option **Dieser Benutzer:**, und geben Sie die Kontodetails des benutzerdefinierten Kontos ein, das Sie im Vorfeld erstellt haben.

    Auf diese Weise konfigurieren Sie die COM+-Anwendung für die Ausführung unter Verwendung des benutzerdefinierten Kontos.
16. Klicken Sie auf **OK**, um das Dialogfeld **Eigenschaften** zu schließen.
17. Schließen Sie die Komponentendienste.

9. Entwickeln einer Webanwendung zum Aufrufen der Serviced Component

In diesem Verfahren wird eine einfache ASP.NET-Webanwendung entwickelt, die als Clientanwendung zum Aufrufen des Webdienstes (über die Serviced Component) verwendet wird.

▶ **So entwickeln Sie eine Webanwendung zum Aufrufen der Serviced Component**

1. Erstellen Sie auf dem Webdienst-Clientcomputer mit C# eine neue ASP.NET-Webanwendung mit Namen **SecureMathClient**.
2. Fügen Sie einen Verweis auf **System.EnterpriseServices** hinzu.
3. Fügen Sie einen Verweis auf die Serviced Component **WebServiceRequestor** hinzu.

   Suchen Sie die Datei **WebServiceRequestor.dll**, die sich im Projektverzeichnis **WebServiceRequestor** im Ordner **bin\debug** befindet.
4. Öffnen Sie **WebForm1.aspx.cs**, und fügen Sie unterhalb der vorhandenen **using**-Anweisungen die folgende **using**-Anweisung hinzu.

```
using WebServiceRequestor;
```

5 Zeigen Sie **WebForm1.aspx** im Entwurfsmodus an, und erstellen Sie das in Abbildung 2 gezeigte Formular unter Verwendung der folgenden IDs:

- operand1
- operand2
- result
- add

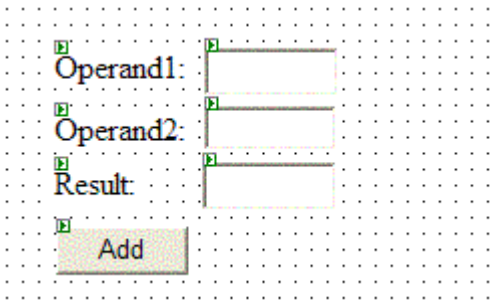

Abbildung 2

Anordnung der Steuerelemente im Webformular

6 Doppelklicken Sie auf **Add**, um einen Ereignishandler für das Klickereignis der Schaltfläche zu erstellen.

7. Fügen Sie dem Ereignishandler den folgenden Code hinzu.

Hinweis: Legen Sie die Zeichenfolge **certPath** auf die Speicherposition der Zertifikatdatei fest, die Sie in Verfahren 6, "Exportieren des Clientzertifikats in eine Datei", exportiert haben.

Legen Sie die Zeichenfolge **url** auf den HTTPS-URL Ihres Webdienstes fest.

```
private void add_Click(object sender, System.EventArgs e)
{
  // TODO: Replace with a valid path to your certificate
  string certPath = @"C:\CustomAccountCert.cer";
  // TODO: Replace with a valid URL to your Web service
  string url = "https://wsserver/securemath/math.asmx";
  MathServiceComponent mathComp = new MathServiceComponent();

  long addResult = mathComp.CallMathWebService( certPath,
                                                url,
                                                Int32.Parse(operand1.Text),
                                                Int32.Parse(operand2.Text));
  result.Text = addResult.ToString();
}
```

8. Klicken Sie im Menü **Erstellen** auf **Projektmappe erstellen**.
9. Führen Sie die Anwendung aus. Geben Sie zwei Zahlen ein, die addiert werden sollen, und klicken Sie dann auf **Add**.

   Die Webanwendung ruft nun die Serviced Component auf, die ihrerseits wiederum den Webdienst unter Verwendung von SSL aufruft und das Clientzertifikat übergibt.

Weitere Ressourcen

Weitere Informationen finden Sie unter "Vorgehensweise: Einrichten von SSL auf einem Webserver" im Abschnitt "Referenz" dieses Handbuchs.